AFIT/GAM/ENC/03S-1

STOCHASTIC INTRA-CELLULAR MODELING

THESIS

Thomas Hopkins
Civilian

AFIT/GAM/ENC/03S-1

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

Research sponsored in part by the Air Force Research Laboratory, Air Force Materiel Command, USAF. The United States Government is authorized to reproduce and distribute reprints notwithstanding any copyright notation thereon. The views and conclusions contained in this thesis are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory, Department of Defense, or the United States Government.

AFIT/GAM/ENC/03S-1

STOCHASTIC INTRA-CELLULAR MODELING

THESIS

Presented to the Faculty

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Applied Mathematics

Thomas Hopkins, B.S.

Civilian

September, 2003

AFIT/GAM/ENC/03S-1

STOCHASTIC INTRA-CELLULAR MODELING

THESIS

Thomas Hopkins, B.S.

Civilian

Approved:

//signed//	
Dr. Dennis Quinn Thesis Advisor	Date
//signed//	
Lt. Col. Lawrence Chilton Committee Member	Date
//signed//	
Dr. William Baker Committee Member	Date

Acknowledgements

I would like to thank my committee members Dr. Quinn, Dr. Baker, and Lt. Col. Chilton for their oversight. I would especially like to thank Dr. Quinn since this thesis and the work described herein would not have been possible without his expertise and collaboration.

I want to thank Landon Locke, a summer intern for Dr. Quinn, for his work on the UNC software.

I want to acknowledge the work done by Matt Campbell and Jack Young whose work I followed upon. I used their respective thesis's to learn the deterministic approaches used for modeling systems of chemical reactions.

I want to thank my parents and family for their support.

I want to recognize Maria Tjani, an undergraduate math professor at the University of Arkansas at Fayetteville. She was a really great teacher and caused me to look more closely at studying math.

I would also like to thank Gary Gygax and Dave Arneson, creators of the game Dungeons & Dragons, for the many joy filled hours of stress relief that were very necessary during this arduous process.

Thomas Hopkins

Table of Contents

Page

List of Figures

List of Tables

Abstract

Air Force personnel may sometimes come into contact with potentially harmful chemicals while performing their duties. Of course the Air Force desires to keep any potential health risks to its members to a minimum. To this end the Air Force would like to identify which chemicals are toxic, their level of toxicity, and the processes by which these chemicals disrupt normal biological activities at the cellular level. The development of mathematical models can be of great benefit to toxicity studies.

One area in which mathematical modeling can be used is to further the understanding of the intra-cellular processes by which a healthy cell and a poisoned cell behave. If a complete mathematical model could be constructed that described the inner workings of a living cell, then perhaps the introduction of potentially toxic chemicals could be analyzed and the resulting cell events could be better understood. Mathematical modeling is not a replacement for wet lab experiments but can be used in conjunction with lab experimentation. Because real world systems involve randomness, that is noise, and the desire is to create mathematical models to represent those systems, it is necessary to study approaches used to add noise to mathematical models.

This document examines different methods for incorporating noise into biochemical systems. The various quantities involved in the reactions are treated as random variables. The methods can be separated into two categories: those which treat the random variable as having a continuous state space and those which treat the random variable as having a discrete state space. The use of these different approaches are compared in order to better understand what type of method would be best used for adding noise to a model and how the model is affected.

It is hoped that this work is a useful step towards the Air Force's understanding of modeling intra-cellular processes.

STOCHASTIC INTRA-CELLULAR MODELING

I. Introduction

1.1 Overview

Air Force personnel may sometimes come into contact with potentially harmful chemicals while performing their duties. Of course the Air Force desires to keep any potential health risks to its members to a minimum. To this end the Air Force would like to identify which chemicals are toxic, their level of toxicity, and the processes by which these chemicals disrupt normal biological activities at the cellular level. An example is the chemical hydrazine. The Human Effectiveness Directorate funded research utilizing Affymetrix gene chips to study the toxicity effects of hydrazine. The development of mathematical models can be of great benefit to toxicity studies.

One area in which mathematical modeling comes into use in reaching the Air Force's goals is with regards to the understanding of the intra-cellular processes by which a healthy cell and a poisoned cell behave. If a complete mathematical model could be constructed that described the inner workings of a living cell, then perhaps the introduction of potentially toxic chemicals could be analyzed and the resulting cell events could be better understood. Mathematical modeling is not a replacement for wet lab experiments but can be used in conjunction with lab experimentation. Because real world systems involve randomness, that is noise, and the desire is to create mathematical models to represent those systems, it is necessary to study approaches used to add noise to mathematical models.

This document examines different methods for incorporating noise into bio-chemical systems. The amounts of the species involved in the reactions are treated as random variables. The methods can be separated into two categories: those which

treat the random variable as having a continuous state space and those which treat the random variable as having a discrete state space. Stochastic differential equations are solved for the continuous approaches. The use of four BioSPICE modules are studied: Stochastica, which was developed at UCLA; Exact Stochastic Simulator (ESS), which was developed at the University of Tennessee; Gene Regulatory Analysis and Stochastic Simulation (GRASS), which was developed at Boston University; SDEsolver, which was developed at the University of North Carolina. Stochastica and ESS treat the state space of the species' random variables as being discrete, while GRASS treats the state space as being continuous. The University of North Carolina module uses a hybrid method that involves both the continuous and discrete approach. Six algorithms are examined and implemented in Matlab. Four of these are methods for solving stochastic differential equations: the explicit Euler-Maruyama method, implicit Euler-Maruyama method, the explicit Milstein method, and the implicit Milstein method. The other two, the exact stochastic simulation of Gillespie and Gillespie's τ-leaping method treat the state space as being discrete. The use of these different methods and software are compared in order to better understand what type of method would be best used for adding noise to a model and how that model is affected.

1.2 Organization of Document

This document is organized as follows:

Chapter 2 briefly discusses discrete and continuous deterministic approaches to modeling of chemical reactions. The discrete technique mentioned is a Boolean approach. The continuous modeling is done using the rate equation approach and three different software programs that employ it are discussed: Gepasi, BioCharon, and JigCell. Also a hybrid method involving a discrete and continuous approach is mentioned. The motivation for adding noise to chemical systems is presented. The use of stochastic differential equations is discussed along with a conventional

approach used to solve them. The Fokker-Planck equation is introduced. Gillespie's exact method and the software package BioSPICE are also introduced.

Chapter 3 describes the approaches and software programs used for continuous and discrete stochastic modeling of chemical reactions. The continuous modeling approaches involve stochastic differential equations. The different methods discussed and then used to analyze these are: the Euler-Maruyama method, the Milstein method, the Fokker-Planck equation, and a variable step size integration method. The BioSPICE module GRASS is also examined. The discrete modeling approaches involve the exact method, the plain τ-leaping method, and the estimated-midpoint τ-leaping method. Also, the BioSPICE modules Stochastica and Exact Stochastic Simulator, which employ the exact method, are examined. The University of North Carolina module in BioSPICE, which performs continuous, discrete, and a hybrid method involving both continuous and discrete modeling is examined.

Chapter 4 presents and discusses the results of the experiments described in the procedures in chapter 3.

II. Background

2.1 Overview

The analysis of biological processes provides a rich field of study. Two major areas are modeling of biological systems and data analysis. Metabolic pathways, gene regulation, and other cell transport systems are biological processes that can and have been modeled. Mathematical modeling provides numerous benefits to the biologists. Wet lab experiments can be very expensive. Mathematical models provide inexpensive analysis of a process. Modeling is also faster since many more replications can be done than during the same period of time in a lab and it also allows for easy control and manipulation of the system's parameters. Modeling can also be used as an exploratory tool before experiments are performed. This could allow lab experiments to be designed with insight into the possible behaviors of the system. Of course, real world experiments will always be necessary but the use of mathematical models and analysis can be a useful and sometimes necessary tool to understanding different biological processes. The use of mathematical algorithms in data analysis is essential. Many biological experiments, such as gene micro-array expression experiments, create huge amounts of data. It would be almost impossible to extract information without some kind of mathematical analysis. Mathematicians have developed many different methods that can be applied to understand biological phenomena. Some of the different modeling methods will be discussed in this chapter.

2.2 Deterministic Methods

Deterministic approaches allow for the state of the system to be known at any time for any given set of parameter values. There are a variety of deterministic approaches that have been created. Three approaches mentioned here are: the Boolean approach, the rate equation approach, and the Boolean-continuous approach.

2.2.1 Boolean Approach. This approach uses Boolean logic to model biological systems. The system being analyzed is considered analogous to an electric circuit. For example, if modeling the production of proteins, the genes are either considered to be in the on or off position at any given time, similar to an electric switch. "An example of such a rule is if genes A and B were ON at the current timestep and gene C was OFF, only then is gene D ON at the current timestep" [30, pg 250]. Smolen et al. state that the usefulness of Boolean networks is in their ability to model feedback loops. Feedback loops are found in genetic systems, "...negative feedback loops are, quite generally, important for maintaining homeostasis in levels of gene products, and that positive feedback loops are important for allowing multiple stable states of gene product levels..." [30, pg 251].

Smolen et al. point out drawbacks of the Boolean approach in comparison to a continuous approach, i.e. the use of differential equations. The Boolean models do not always have the same steady states as the continuous approach. This is a problem because the continuous approach is considered to be a more accurate representation of physical processes. Also, the Boolean models sometimes contain periodic behavior that is not found in the continuous approach [30, pgs 256-257].

2.2.2 Rate Equation Approach. Because biological systems are dynamic in nature this immediately brings to mind the use of differential equations. The rate equation approach involves constructing differential equations from the stoichiometric chemical equations. Detailed explanations and examples can be found in references [19], [7], and [34]. Several different software packages have been developed to implement this approach: Gepasi, BioCharon, and JigCell are examples. The common theme of these programs is to provide a simple and straightforward way of examining the system of reactions without directly creating and solving a system of differential equations. Interesting types of system behavior have been examined using differential equations, such as bistability and limit cycles. Gepasi, BioCharon, and Jigcell can be used to investigate these properties.

2.2.2.1 Gepasi. Gepasi is free software that was developed by Mendes [12]. The chemical reactions are input by the user in stoichiometric form: A + B $\rightarrow$ C. These are then translated into a system of differential equations and solved. The program performs a wide variety of analysis techniques including various optimization techniques [27]. Gepasi has the capability to graph 2D and 3D plots of the concentration levels throughout the specified time period. The use of this software was examined by Campbell [7]. Mendes is now working on a new software program that will be called COPASI. It will perform the same functions as Gepasi but it will also carry out stochastic integrations.

2.2.2.2 BioCharon. BioCharon is a program that is included in a larger collection of software programs called BioSPICE [3]. It consists of two modules: Bio Sketch Pad and Charon. Bio Sketch Pad is a graphic user interface that allows the user to build a wiring diagram to represent the system of chemical reactions. Nodes are used to represent the different species involved. Nodes that are connected using directed arrows indicate the reactions. For example, a directed arrow that points from node A to node B represents the chemical reaction in which species A is being consumed and species B is being created. More complex systems involving multiple chemical reactions can be modeled. The graphic representation is then translated into a system of differential equations. Charon solves this system of differential equations and provides graphic output of the different species concentrations over the defined time period. The use of this software was examined by Young [34].

2.2.2.3 JigCell. JigCell is another program that is a part of the BioSPICE package. JigCell consists of three subprograms: JigCell Model Builder, JigCell Run Manager, and Comparator [22]. The user interface is in a spreadsheet format. Using the Model Builder, the user enters the chemical reactions in stoichiometric form and specifies the rate constants. The default rate-law is mass

action [34, pg 2-12] but the program allows the user to define their own rate-law equations. Run Manager controls the simulation runs. Here is where the user can change parameter values and species' initial concentrations. The comparator is used for analyzing and comparing experimental data to results from mathematical models. The use of this software was examined by Young [34].

2.2.3 Boolean-Continuous Approach. A hybrid approach uses aspects of both the Boolean and rate equation approaches. This method was used successfully by McAdams and Shapiro [25] to study genetic regulation, the lysis-lysogeny pathway, with regard to the bacteriophage λ. The bacteriophage is a virus that infects bacteria cells [32]. The regulation of two different proteins determine whether bacteriophage λ becomes dormant in the DNA, this is lysogeny, of the bacteria or whether it reproduces itself and lyses the cell. McAdams and Shapiro used Boolean switches to incorporate time delays and pathway changes in the model and solved differential equations to find protein concentrations.

2.3 Noise in Biological Models

Frequently a deterministic approach can very accurately represent the behavior exhibited by the system being studied. However, the deterministic approach does not take into account the naturally occurring randomness in biological processes. Many of these random behaviors are not only present and very influential on these processes but they may be considered desirable for the organism in which they are taking place. Bacterial and yeast cells "...can exploit noise in some developmental switches to deliberately introduce indeterminism into the switching and randomize phenotypic outcomes" [26, pg 65].

Gardner et al. performed a lab experiment involving gene regulation using E. coli [11]. They created a synthetic genetic toggle switch that consisted of two repressors and two promoters. "Each promoter is inhibited by the repressor that is

transcribed by the opposing promoter" [11, pg 339]. Depending on which promoter was being expressed, the system would be defined to be in the high or low state. They created differential equations to describe the behavior of the system. The experimental results differed from those obtained using the deterministic model. "Owing to the natural fluctuations in gene expression, the bifurcation is not a perfect discontinuity as predicted by the deterministic toggle equations" [11, pg 341].

Hasty et al. examined a system of chemical reactions that model the lysis-lysogeny pathway for the bacteriophage λ [18]. Like the Gardner et al. experiment Hasty et al. analyzed the switching behavior of the pathway. Specifically they studied the use of external noise as the switching mechanism. Hasty et al. also examined a smaller set of chemical reactions that describes the regulation of the production of a generic protein [17]. This smaller set of reactions is contained in the lysis-lysogeny system of reactions. The smaller model will be used for the analysis throughout this document. It is represented by the following set of chemical reactions:

$$\xrightarrow{r} \quad X \tag{2.1}$$

$$2X \quad \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} \quad X_2 \tag{2.2}$$

$$D \ + \ X_2 \quad \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} \quad DX_2 \tag{2.3}$$

$$DX_2 \ + \ P \quad \xrightarrow{k_3} \quad DX_2 \ + \ P \ + \ nX \tag{2.4}$$

$$X \quad \xrightarrow{k_4} \tag{2.5}$$

Equation (2.1) represents "the basal rate of production of protein, i.e., the low baseline expression rate in the absence of a transcription factor" [17, pg 193]. Equation (2.2) represents the dimerization of the protein X. Equation (2.3) represents the formation of the DNA-promoter complex at site D. Equation (2.4) represents

transcription and translation for protein X. P is the RNA polymerase required for transcription. Equation (2.5) represents the degradation of protein X.

Using the rate equation approach, a system of differential equations has been derived. The derivations are not shown here since Campbell presents them in great detail [7, pgs 3-17 - 3-20]. The reactions give the following system of differential equations:

$$\frac{d[X]}{dt} = r + 2k_{-1}[X_2] - 2k_1[X]^2 + nk_3[DX_2][P] \tag{2.6}$$

$$\frac{d[X_2]}{dt} = k_1[X]^2 - k_{-1}[X_2] - k_2[D][X_2] + k_{-2}[DX_2] \tag{2.7}$$

$$\frac{d[D]}{dt} = k_{-2}[DX_2] - k_2[D][X_2] \tag{2.8}$$

$$\frac{d[DX_2]}{dt} = k_2[D][X_2] - k_{-2}[DX_2] \tag{2.9}$$

where the brackets denote concentration of the species. If the assumption is made that reactions (2.4) and (2.5) are much slower than the others, then the system of differential equations can be reduced to one differential equation. This is because the only species that has a change in concentration as a result of reactions (2.4) and (2.5) is X. Thus, differential equations (2.7), (2.8), and (2.9) are assumed to reach steady state much quicker than equation (2.6). Campbell presents a well detailed derivation [7, pgs 3-20 - 3-23]. The resulting single differential equation is of the form:

$$\frac{dX}{dt} = \frac{\alpha X^2}{1+\beta X^2} - \gamma X + r \tag{2.10}$$

where

$$\alpha = nk_3pK_1K_2d_T, \qquad \beta = K_1K_2, \qquad \gamma = k_4, \qquad K_1 = \frac{k_1}{k_{-1}}, \qquad K_2 = \frac{k_2}{k_{-2}},$$

and d_T is the total concentration of DNA binding sites, ie. the sum of D and DX_2. By examining the concentration values of X that cause $\frac{dX}{dt}$ to be equal to zero in equation (2.10), the different steady state concentrations of X can be found. This causes the right hand side of equation (2.10) to equal zero which yields a third order polynomial whose roots are referred to as critical values [6, pgs 460,474]. These critical values are candidates for being possible stable steady state values. Sometimes all three roots are real and sometimes only one root is real. When there are three real roots, the system has a bifurcation point [6, pgs 73,115]. The critical values of X can be determined in terms of α. A bifurcation plot is presented in Figure 2.1 [17, pg 193]. The dotted line represents unstable critical values.

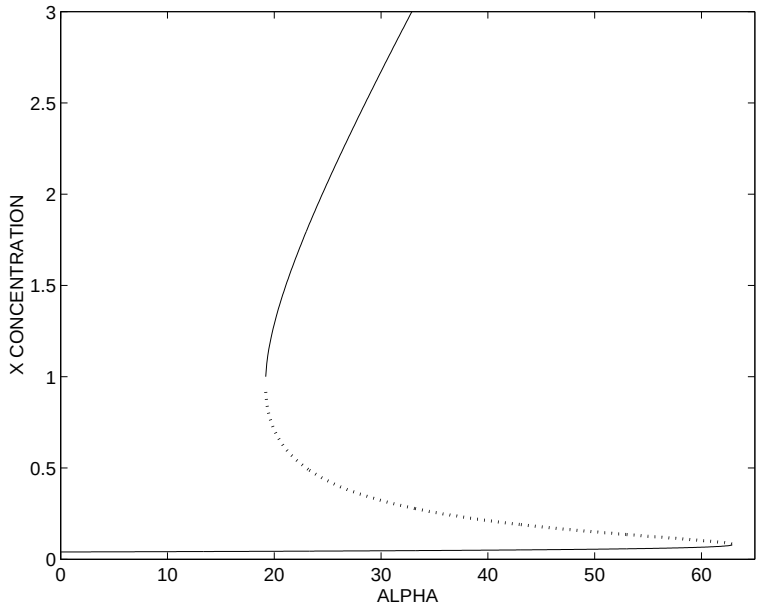


Figure 2.1: Bifurcation Plot for Protein Regulation Model

For the parameters used by Hasty et al., there were two stable critical values separated by an unstable critical value. Using equation (2.10), if the initial X concentration was $+\varepsilon$ or $-\varepsilon$ from the middle unstable critical value, then the concentration would eventually end up at the upper stable critical value or the lower stable critical value respectively [17, pg 193]. Note that when examining the full system of differential equations this may not necessarily be the case. This is because the crit-

ical values are on a four dimensional surface. The other species concentrations have an effect on which critical value the full system ends up at. Hasty et al. examined adding increasingly stronger noise to equation (2.10) and found that the stronger the noise was, the more likely that the X concentration would be attracted to the larger stable critical value.

Many of the techniques to be described later refer to a Markov process. Therefore it is important to understand the Markov property. The Markov property is defined as [23, pg 17]:

$$P\{X(t_n) = x_n | X(t_{n-1}) = x_{n-1}, X(t_{n-2}) = x_{n-2}, ..., X(t_0) = x_0\} \;=\; P\{X(t_n) = x_n | X(t_{n-1}) = x_{n-1}\} \tag{2.11}$$

That is, the conditional probability of the random variable X being in a certain state at time t is only dependent on the state from which X is transitioning. This probability is independent of all other previous values X might have been in. This property is appropriate for describing chemical reactions. The chemical reactions that are about to take place are only dependent on the concentrations of the chemicals at that time.

2.3.1 Stochastic Differential Equations. Stochastic differential equations (SDE) provide a natural way to introduce randomness into a system of rate equations. Suppose the change in concentration of a species X is represented by the following rate equation:

$$\frac{dX(t)}{dt} \;=\; f(X(t)) \tag{2.12}$$

To represent the effect of noise on the concentration , a random variable $\xi(t) \sim N(0,1)$ is added to the differential equation, where $\eta(t) \sim N(\mu, \sigma^2)$ indicates that

$\eta(t)$ is a normally distributed random variable with mean μ and variance σ^2 [33, pgs 170-171]. This notation will be used throughout the document. The resulting stochastic differential equation is of the form:

$$\frac{dX(t)}{dt} \quad = \quad f(X(t)) \; + \; g(X(t))\xi(t) \tag{2.13}$$

If g(X(t)) is equal to a constant and does not depend on X, though it could still depend on t, this is referred to as additive or white noise. If g(X(t)) depends on X this would be called multiplicative or colored noise [24, pg 118]. Integrating SDE (2.13) results in

$$X(t) \quad = \quad X(t_0) \; + \; \int_{t_0}^{t} f(X(\tau))\, d\tau \; + \; \int_{t_0}^{t} g(X(\tau))\xi(\tau)\, d\tau \tag{2.14}$$

If this were a deterministic equation, Riemann integration could be used to readily find the solution. However, Riemann integration does not make sense for the integral involving $\xi(t)$. This can be seen from the following theorem [1, pg 171].

Theorem

Let f be defined and bounded on [a, b] and let D denote the set of discontinuities of f in [a, b]. Then f $\epsilon \, \mathbb{R}$ *(reals) on [a, b] if, and only if, D has measure zero.*

For a definition of a set of measure zero see [1, pg 169].

The first problem is that $\xi(t)$ is not a bounded function over the interval of integration. This is not that severe of a handicap because the probability of getting values more than a few standard deviations away from the mean quickly drops to effectively being zero. The random variable $\xi(t)$ could be redefined so that values greater than the mean plus three standard deviations are set equal to the mean plus three standard deviations. Values less than the mean minus three standard deviations are set equal to the mean minus three standard deviations. This would still include 99.73% of the values that $\xi(t)$ would take on. Even the redefined $\xi(t)$

is not Riemann-integrable. This is because $\xi(t)$ is everywhere discontinuous. It's set of discontinuities is not a set of measure zero.

The integral of $\xi(t)$ is not Lebesgue integrable either. The problem is again that the function is everywhere discontinuous. A common example of a function that is everywhere discontinuous but is still Lebesgue integrable is the function over the interval [0,1] which has the value 0 at every irrational number and 1 at every rational number. Even though this function is discontinuous at every point it can be looked at as being equal to the zero function except on a set of measure zero. This is not the case for $\xi(t)$. It is not almost everywhere equal to a Lebesgue integrable function.

A standard approach for dealing with SDE (2.13) is to consider the random variable $\xi(t)$ to be the derivative of a Wiener random variable. It should be noted that this is only taken to be true in a formal sense. This is because a Wiener random variable is nowhere differentiable. The Wiener process, sometimes also called a Brownian process, is a Markov process that describes transitions of a random variable over small intervals of time. It is a mathematical structure that was created to describe "the erratic motion of a grain of pollen on a water surface due to its continually being bombarded by water molecules" [24, pg 40]. This type of random motion is referred to as Brownian motion. The following are the properties of a Wiener process $\{W(t), t \geq 0\}$ [21, pg 148]:

1. W(t) has independent time increments. That is, for every pair of disjoint time intervals $(t_1, t_2), (s_1, s_2)$ the random variables $W(t_2) - W(t_1)$ and $W(s_2) - W(s_1)$ are independent.
2. $W(t_2) - W(t_1) \sim N((t_2 - t_1)\mu, (t_2 - t_1)\sigma^2)$

Substituting $\frac{\partial W(\tau)}{\partial \tau}$ for $\xi(\tau)$ in equation (2.14) yields:

$$X(t_{i+1}) \quad = \quad X(t_i) \; + \; \int_{t_i}^{t_{i+1}} f(X(\tau))d\tau \; + \; \int_{t_i}^{t_{i+1}} g(X(\tau))dW(\tau) \qquad (2.15)$$

Equation (2.15) can be discretized using the forward rectangular rule to obtain:

$$X(t_{i+1}) \quad = \quad X(t_i) \; + \; f(X(t_i))(t_{i+1} - t_i) \; + \; g(X(t_i))(W(t_{i+1}) - W(t_i)) \quad (2.16)$$

When g(X(t)) is evaluated at the left hand point of the interval [i,i+1] as done in equation (2.16), this is known as the Ito integral. There is also a form where g(X(t)) is evaluated at the midpoint known as the Stratonovich integral. There is not a special name given when g(X(t)) is evaluated at the right hand end point.

SDE (2.13) is now in the form of the explicit Euler-Maruyama (EM) method in equation (2.16) [16, pg 277] [24, pg 305]. Using the backward rectangular rule on equation (2.15) would result in the implicit EM method [24, pg 396]. The EM method is one of the most common and simplest methods used to solve stochastic differential equations. Higham's paper [20] provides an excellent introduction for the novice to numerical calculations of Brownian motion paths, the explicit Euler-Maruyama and explicit Milstein methods, and strong and weak convergence of the just mentioned methods.

A method has strong order of convergence [16, pgs 276-277] [20, pg 534] of γ if there exists a constant C such that

$$\mathbf{E} \mid X_L - X(T) \mid \quad \leq \quad C(\Delta t)^{\gamma} \qquad (2.17)$$

while a method has weak order of convergence [16, pgs 276-277] [20, pg 537] of γ if there exists a constant C such that

$$| \mathbf{E}[X_L] - \mathbf{E}[X(T)] | \quad \leq \quad C(\Delta t)^{\gamma} \tag{2.18}$$

where LΔt = T and $\mathbf{E}$ is the expected value. Thus X(T) is the true value and X_L is the method approximation at time T.

2.3.2 Fokker-Planck Equation. The Fokker-Planck equation, which is also known as the forward Kolmorogov equation, is a partial differential equation which is satisfied by the probability density function of a Markov process in which the state space of the random variable is a vector in Euclidean n space [2, pgs 129,133]. This type of process is also called a diffusion process. The derivation of the Fokker-Planck is presented here for completeness [21, pgs 172-174].

Let $R(\vec{x})$ be a function whose first two derivatives vanish on the boundary of a set I ϵ $\mathbb{R}^n$. The arrow over x indicates that it is a vector in $\mathbb{R}^n$ and $p(\vec{x}, t)$ denotes a probability density function.

$$\begin{aligned} \int_I R(\vec{x}) \frac{\partial p(\vec{x}, t)}{\partial t} d\vec{x} \quad &= \quad \int_I R(\vec{x}) \lim_{\tau \to 0} \left[\frac{p(\vec{x}, t+\tau) - p(\vec{x}, t)}{\tau} \right] d\vec{x} \\ &= \quad \lim_{\tau \to 0} \frac{1}{\tau} \int_I R(\vec{x}) \left[p(\vec{x}, t+\tau) - p(\vec{x}, t) \right] d\vec{x} \end{aligned} \tag{2.19}$$

Now the law of total probability is used to rewrite $p(\vec{x}, t+\tau)$ [33, pg 68]:

$$\int_I p(\vec{x}, t+\tau \mid \vec{x_0}, t) p(\vec{x_0}, t) d\vec{x_0} \;=\; p(\vec{x}, t+\tau) \tag{2.20}$$

Note that $p(\vec{x}, t+\tau \mid \vec{x_0}, t)$ indicates the conditional probability that the system is in state $\vec{x}$ at time $t+\tau$ given that it was in state $\vec{x_0}$ at time t. Substituting equation (2.20) into equation (2.19) yields:

$$\int_I R(\vec{x})\frac{\partial p(\vec{x},t)}{\partial t}d\vec{x} \quad = \quad \lim_{\tau\to 0}\frac{1}{\tau}\int_I R(\vec{x})\left[\int_I p(\vec{x},t+\tau \mid \vec{x_0},t)p(\vec{x_0},t)d\vec{x_0} - p(\vec{x},t)\right]d\vec{x}$$

$$= \quad \lim_{\tau\to 0}\frac{1}{\tau}\left[\int_I\int_I R(\vec{x})p(\vec{x},t+\tau \mid \vec{x_0},t)p(\vec{x_0},t)d\vec{x_0}d\vec{x} - \int_I R(\vec{x})p(\vec{x},t)d\vec{x}\right]$$

$R(\vec{x})$ in the double integral is now expanded using the Taylor series.

$$\int_I R(\vec{x})\frac{\partial p(\vec{x},t)}{\partial t}d\vec{x} \quad = \quad \lim_{\tau\to 0}\frac{1}{\tau}\Bigg[\int_I\int_I\Bigg[R(\vec{x_0}) + \sum_i (x-x_0)_i\frac{\partial R(\vec{x_0})}{\partial x_{0i}}$$

$$+\frac{1}{2}\sum_j\sum_i (x-x_0)_j(x-x_0)_i\frac{\partial^2 R(\vec{x_0})}{\partial x_{0j}x_{0i}}+\ldots\Bigg]p(\vec{x},t+\tau \mid \vec{x_0},t)p(\vec{x_0},t)d\vec{x_0}d\vec{x} - \int_I R(\vec{x})p(\vec{x},t)d\vec{x}\Bigg]$$

$$= \quad \lim_{\tau\to 0}\frac{1}{\tau}\Bigg[\int_I\int_I R(\vec{x_0})p(\vec{x},t+\tau \mid \vec{x_0},t)p(\vec{x_0},t)d\vec{x}d\vec{x_0}$$

$$+\int_I\int_I\sum_i (x-x_0)_i\frac{\partial R(\vec{x_0})}{\partial x_{0i}}p(\vec{x},t+\tau \mid \vec{x_0},t)p(\vec{x_0},t)d\vec{x}d\vec{x_0}$$

$$+\int_I\int_I\frac{1}{2}\sum_j\sum_i (x-x_0)_j(x-x_0)_i\frac{\partial^2 R(\vec{x_0})}{\partial x_{0j}x_{0i}}p(\vec{x},t+\tau \mid \vec{x_0},t)p(\vec{x_0},t)d\vec{x}d\vec{x_0}$$

$$+ \int_I\int_I h.o.t.\, d\vec{x}d\vec{x_0} \; - \; \int_I R(\vec{x})p(\vec{x},t)d\vec{x}\Bigg]$$

where h.o.t. represents the higher order terms of the Taylor series. The first term and the last term cancel since: $\int_I p(\vec{x},t+\tau \mid \vec{x_0},t)d\vec{x} \; = \; 1$. The reason for this is because the sum of the probabilities of transitioning from $\vec{x_0}$ to every other possible value must equal to one. Thus

$$\int_I R(\vec{x})\frac{\partial p(\vec{x},t)}{\partial t}d\vec{x} \quad = \quad \lim_{\tau\to 0}\frac{1}{\tau}\Bigg[\int_I\int_I\sum_i(x-x_0)_i\frac{\partial R(\vec{x_0})}{\partial x_{0i}}p(\vec{x},t+\tau \mid \vec{x_0},t)d\vec{x}p(\vec{x_0},t)d\vec{x_0}$$

$$+\int_I\int_I\frac{1}{2}\sum_j\sum_i(x-x_0)_j(x-x_0)_i\frac{\partial^2 R(\vec{x_0})}{\partial x_{0j}x_{0i}}p(\vec{x},t+\tau \mid \vec{x_0},t)d\vec{x}p(\vec{x_0},t)d\vec{x_0}+\int_I\int_I h.o.t.\,d\vec{x}d\vec{x_0}\Bigg]$$

Using "...requirements on the short-time properties of the transition probability..." [21, pg 172] that define diffusion processes, the inner integrals are now reduced to the following forms:

$$\int_I(x-x_0)_i p(\vec{x},t+\tau \mid \vec{x_0},t)d\vec{x} \quad = \quad A_i(x_0,t)\tau+o(\tau) \tag{2.21}$$

$$\int_I\frac{1}{2}(x-x_0)_i(x-x_0)_j p(\vec{x},t+\tau \mid \vec{x_0},t)d\vec{x} \quad = \quad D_{ij}(x_0,t)\tau+o(\tau) \tag{2.22}$$

$$\int_I h.o.t.\,d\vec{x} \quad = \quad o(\tau) \tag{2.23}$$

where $o(\tau)$ has the property that as τ approaches zero, $\frac{o(\tau)}{\tau}$ approaches zero. Equations (2.21) and (2.22) represent the "infinitesimal mean and variance of the change in x(t), respectively" [2, pg 132]. Taking the limit as τ goes to zero results in the following:

$$\int_I R(\vec{x})\frac{\partial p(\vec{x},t)}{\partial t}d\vec{x} \quad = \quad \int_I\sum_i\left[A_i(x_0,t)p(\vec{x_0},t)\frac{\partial R(\vec{x_0})}{\partial x_{0i}}\right]d\vec{x_0}+\int_I\sum_j\sum_i\left[D_{ij}(x_0,t)p(\vec{x_0},t)\frac{\partial^2 R(\vec{x_0})}{\partial x_{0j}x_{0i}}\right]d\vec{x_0}$$

Integrating by parts yields the Fokker-Planck equation.

$$\frac{\partial p(\vec{x},t)}{\partial t} \quad = \quad -\sum_i\frac{\partial[A_i(\vec{x},t)p(\vec{x},t)]}{\partial x_i} \quad + \quad \frac{1}{2}\sum_j\sum_i\frac{\partial^2[D_{ik}(\vec{x},t)p(\vec{x},t)]}{\partial x_j\partial x_i} \tag{2.24}$$

A Fokker-Planck representation for an n dimensional system of stochastic differential equations can be written using equation (2.24). The entries of A(x,t) are the expected values of the derivatives of $x_1, x_2, ..., x_n$. D(x,t) is the variance-covariance matrix of the derivatives of $x_1, x_2, ..., x_n$ [28, pgs 194-196]. As an example, the A(x,t) vector and D(x,t) matrix for the Fokker-Planck representation are given for the following system of stochastic differential equations:

$$\frac{dx_1}{dt} = f_1(\vec{x}) + g_1(\vec{x})\,\xi_1(t) \tag{2.25}$$

$$\frac{dx_2}{dt} = f_2(\vec{x}) + g_2(\vec{x})\,\xi_2(t) \tag{2.26}$$

where $\xi_1(t) \sim N(\mu_1, \sigma_1^2)$ and $\xi_2(t) \sim N(\mu_2, \sigma_2^2)$. The resulting A(x,t) vector and D(x,t) matrix are:

$$A(x,t) = \begin{pmatrix} f_1(\vec{x}) + g_1(\vec{x})\mu_1 \\ f_2(\vec{x}) + g_2(\vec{x})\mu_2 \end{pmatrix} \qquad D(x,t) = \begin{pmatrix} g_1(\vec{x})^2\sigma_1^2 & Cov\{\frac{dx_1}{dt}, \frac{dx_2}{dt}\} \\ Cov\{\frac{dx_2}{dt}, \frac{dx_1}{dt}\} & g_2(\vec{x})^2\sigma_2^2 \end{pmatrix}$$

where $Cov\{\frac{dx_1}{dt}, \frac{dx_2}{dt}\}$ is the covariance of $\frac{dx_1}{dt}$ and $\frac{dx_2}{dt}$ [33, pgs 249-252].

2.3.3 Gillespie's Exact Method. Daniel Gillespie proposed what is known as the exact method [14]. His motivation was to calculate solutions for chemical reactions by looking at the probability of collisions for each molecule. Gillespie demonstrated his algorithm on three different sets of chemical reactions that exhibited oscillatory behavior: the Lotka reactions, the Brusselator, and the Oregonator [14]. The complete theoretical derivation of the method is not presented here. Instead a more basic view is given. The system of chemical reactions is considered as a continuous time Markov chain, or more specifically a Poisson process since the reaction times are assumed to be exponentially distributed [23, pgs 187-231].

Because exponentially distributed random variables are used throughout the justification presented for Gillespie's method, the definitions of the cumulative distribution and probability density functions for X, where X $\sim$ $Exp(\lambda)$, are stated. X $\sim$ $Exp(\lambda)$ indicates that X is an exponentially distributed random variable with mean $\frac{1}{\lambda}$ and variance $(\frac{1}{\lambda})^2$ [33, pgs 178-179]. The cumulative distribution function F(x) is defined as

$$F(x) \;=\; P\{X < x\} \;=\; \left[\begin{array}{cc} 0 & x \;<\; 0 \\ 1 - \exp(-\lambda x) & 0 \;\leq\; x \;\leq\; \infty \end{array}\right.$$

The probability density function f(x) is defined as

$$f(x) \;=\; \left[\begin{array}{cc} 0 & x \;<\; 0 \\ \lambda \exp(-\lambda x) & 0 \;\leq\; x \;\leq\; \infty \end{array}\right.$$

Note that F(x) and f(x) are related by $F(x) = \int_{-\infty}^{x} f(s)\, ds$.

The idea behind Gillespie's method is to treat this as a "...continuous time Markov random walk in the space of all possible [nonnegative] integer populations" [4]. For each chemical reaction a random variable

$$X_i \;\sim\; Exp(a_i) \qquad\qquad i \;=\; 1, 2, ..., n \tag{2.27}$$

is used to represent the time between occurrences of the ith reaction. Thus X_1 represents the amount of time from when reaction 1 last occurred to the next time that reaction 1 takes place.

The goal is to simulate these reactions taking place one at a time. For that reason a new random variable Z is created:

$$Z \quad = \quad min(X_1, X_2, ..., X_n) \tag{2.28}$$

where Z is the amount of time that passes until the very next reaction out of any in the system occurs. Z is also an exponentially distributed random variable as shown in the following proof [23, pg 192]:

$$P\{Z > x\} \quad = \quad P\{min(X_1, X_2, ..., X_n) > x\} \tag{2.29}$$

If the minimum of $X_1, X_2, ..., X_n$ is greater than x, then all of the X_i's are greater than x. Thus

$$P\{Z > x\} \quad = \quad P\{X_1 > x, X_2 > x, ..., X_n > x\} \tag{2.30}$$

The X_i random variables are assumed to be independent, thus

$$P\{Z > x\} \quad = \quad P\{X_1 > x\}P\{X_2 > x\}...P\{X_n > x\} \tag{2.31}$$

The probability statements are written in terms of the complement cumulative distribution function, 1 minus the cumulative distribution function, by making the substitution $P\{X_i > x\} = exp(-\lambda_i x)$. Note that this is just 1 - $P\{X_i < x\}$. This yields

$$\begin{aligned} P\{Z > x\} \quad &= \quad \exp(-\lambda_1 x)\exp(-\lambda_2 x)...\exp(-\lambda_n x) \\ &= \quad \exp(-x(\lambda_1 + \lambda_2 + .. + \lambda_n)) \end{aligned} \tag{2.32}$$

Now the probability for reaction X_1 to occur before any of the others is derived [23, pg 191]. First the law of total probability is used [33, pg 68].

$$P\{X_1 < X_2, X_1 < X_3, ..., X_1 < X_n\} = \int_0^\infty P\{X_1 < X_2, X_1 < X_3, ..., X_1 < X_n \mid X_1 = x\}\lambda_1 \exp(-\lambda_1 x)dx \tag{2.33}$$

Since $P\{X_1 < X_2, X_1 < X_3, ..., X_1 < X_n\} = P\{X_2 > X_1, X_3 > X_1, ..., X_n > X_1\}$, equation (2.33) is now rewritten as

$$P\{X_1 < X_2, X_1 < X_3, ..., X_1 < X_n\} = \int_0^\infty P\{X_2 > X_1, X_3 > X_1, ..., X_n > X_1 \mid X_1 = x\}\lambda_1 \exp(-\lambda_1 x)dx$$

Since $X_1 =$ x, this can now be rewritten as

$$P\{X_1 < X_2, X_1 < X_3, ..., X_1 < X_n\} = \int_0^\infty P\{X_2 > x, X_3 > x, ..., X_n > x\}\lambda_1 \exp(-\lambda_1 x)dx$$

The independence of the X_i random variables yields

$$P\{X_1 < X_2, X_1 < X_3, ..., X_1 < X_n\} = \int_0^\infty P\{X_2 > x\}P\{X_3 > x\}...P\{X_n > x\}\lambda_1 \exp(-\lambda_1 x)dx$$

The probability statements are written in terms of the complement cumulative distribution function by making the substitution $P\{X_i > x\} = exp(-\lambda_i x)$.

$$\begin{aligned} P\{X_1 < X_2, X_1 < X_3, ..., X_1 < X_n\} &= \int_0^\infty \exp(-\lambda_2 x)\exp(-\lambda_3 x)...\exp(-\lambda_n x)\lambda_1 \exp(-\lambda_1 x)dx \\ &= \lambda_1 \int_0^\infty \exp(-x(\lambda_1 + \lambda_2 + ... + \lambda_n))dx \\ &= \frac{\lambda_1}{\lambda_1 + \lambda_2 + ... + \lambda_n} \end{aligned} \tag{2.34}$$

This leaves the exponential rate parameters as the unknowns. These are computed in Gillespie's exact method using the reaction coefficients and the total possible combinations in which the reactant molecules can interact. There are two different implementations of the exact method: the direct method and the first-reaction method.

The direct method algorithm is as follows [13, pgs 415, 417-419]:

1. Set reaction coefficients and initial numbers of molecules for all species. Specify a stopping time and the times to record the molecule values for all species.

2. Using the state vector $\vec{x}^T = (x_1, x_2, ..., x_N)$, calculate $h(\vec{x})_i$, the number of combinations in which the required reactants can interact in order for the ith reaction to occur, using the following formula [33, pg 44]:

$$h(\vec{x})_i = \frac{x_1!}{r_{i,1}!(x_1 - r_{i,1})!}\frac{x_2!}{r_{i,2}!(x_2 - r_{i,2})!} ... \frac{x_N!}{r_{i,N}!(x_N - r_{i,N})!} \qquad i = 1, 2, 3, ..., M \tag{2.35}$$

   where M is the total number of reactions, N is the total number of species, x_j indicates the number of molecules of the jth species, and $r_{i,j}$ indicates the stoichiometric coefficient for the jth species in the ith reaction. Note that the stoichiometric coefficient for a species which does not appear in the ith reaction is equal to zero. Gillespie provides some examples for commonly encountered reactions [13, pgs 405, 413].

3. Calculate

$$a(\vec{x})_i = c_i\, h(\vec{x})_i \qquad i = 1, 2, 3, ..., M \tag{2.36}$$

   where c_i is the reaction rate for the ith reaction, $h(\vec{x})_i$ is described in step 2, and $a(\vec{x})_i$ is the propensity function for the ith reaction. That is when $a(\vec{x})_i$ is multiplied by a suitably small change in time it gives the probability that the ith reaction will occur in the next time step [15, pg 1717]. Equation (2.36) is the same as Gillespie's equation (25) [13, pg 415]

4. Determine the time step:

$$\tau = \frac{1}{a(\vec{x})_1 + a(\vec{x})_2 + ... + a(\vec{x})_M}\, ln\left(\frac{1}{r_1}\right) \tag{2.37}$$

where $r_1 \sim$ Uniform(0,1). Note that $\alpha \sim$ Uniform(a,b) indicates that α is a uniformly distributed random variable on the interval [a,b] [33, pgs 166-168]. Add the time step to the total time passed. Equation (2.37) is the same as Gillespie's equation (27a) [13, pg 420].

5. Determine the reaction that takes place. Calculate $a(\vec{x})_0$, the sum of the propensity terms for each reaction.

$$a(\vec{x})_0 \quad = \quad \sum_{i=1}^{M} a(\vec{x})_i \tag{2.38}$$

First the sum of the $a(\vec{x})_i$ values are multiplied by r_2, where $r_2 \sim$ Uniform(0,1). Then the $a(\vec{x})_i$ values are cumulatively summed up and the first $a(\vec{x})_i$ value that causes the sum to be greater than or equal to $r_2\, a(\vec{x})_0$ means that the ith reaction takes place. Equation (2.38) is the same as Gillespie's equation (26) [13, pg 418].

6. The concentrations are updated according to the reaction that takes place.

7. If the time is greater than or equal to the stopping time quit, otherwise, return to step 2.

The reasoning behind using equation (2.37) to compute the time step τ is now presented. A probability, r_1, for the cumulative distribution is written as:

$$r_1 \quad = \quad 1 - exp(-a(\vec{x})_0 \tau) \tag{2.39}$$

where $r_1 \sim$ Uniform(0,1) and $a(\vec{x})_0$ is defined as in equation (2.38). Rearranging equation (2.39) yields

$$1 - r_1 \quad = \quad exp(-a(\vec{x})_0 \tau) \tag{2.40}$$

Note that 1 - $r_1 = r_2$, where $r_2 \, \epsilon$ [0,1]. Substitution results in

$$r_2 \quad = \quad exp(-a(\vec{x})_0 \tau) \tag{2.41}$$

Taking the natural log of both sides yields

$$ln(r_2) \quad = \quad -a(\vec{x})_0\tau \tag{2.42}$$

τ is now isolated on the left hand side, and this results in equation (2.37):

$$\begin{aligned} \tau &= \frac{ln(r_2)}{-a(\vec{x})_0} \\ &= \frac{1}{a(\vec{x})_0} ln(\frac{1}{r_2}) \end{aligned} \tag{2.43}$$

For the first-reaction method, a time step for each reaction is calculated and the next reaction to occur is the one with the smallest time step. The first-reaction method algorithm is obtained by substituting the following two steps in the direct method algorithm [13, pgs 419-421]:

4 Calculate the amount of time at which each of the reactions will next occur:

$$\tau_i \; = \; \frac{1}{a(\vec{x})_i} ln\left(\frac{1}{r_i}\right) \qquad\qquad i \; = \; 1, 2, 3, ..., M \tag{2.44}$$

where $r_i \sim$ Uniform(0,1). Note that τ_i is not calculated when $a(\vec{x})_i = 0$. Equation (2.44) is the same as Gillespie's equation (29a) [13, pgs 420].

5 The next reaction that occurs is determined by choosing the smallest τ_i.

To better illustrate the use of the exact method algorithm, the form for some of the equations needed for stepping through the algorithm are given for the reactions depicted in equation (2.45) with forward and reverse reaction coefficients of c_f and c_r:

$$A \; + \; B \; \rightleftarrows \; 2C \tag{2.45}$$

The forward reaction is reaction 1 and the backward reaction is reaction 2. The total possible combinations for the interactions of the molecules that lead to the forward and reverse reactions are given by h_1 and h_2 respectively:

$$h_1 = \frac{[A]!}{1!\,([A]-1)!}\frac{[B]!}{1!\,([B]-1)!} = [A]\,[B] \qquad h_2 = \frac{[C]!}{2!\,([C]-2)!} = \frac{[C]\,([C]-1)}{2}$$

where the brackets denote number of molecules of the species. These are then used to determine the propensity functions a_1 and a_2.

$$a_1 = c_f h_1 = c_f [A][B] \qquad a_2 = c_r h_2 = c_r \frac{[C]\,([C]-1)}{2}$$

2.4 BioSPICE

BioSPICE is a collection of software programs [5]. It is a project funded by the Defense Advanced Research Projects Agency (DARPA), which provides funds for research that is of interest to the Department of Defense. The various module programs that make up BioSPICE focus specifically on modeling and analyzing data from biological processes. The goal is for it to "be a user-friendly simulation tool, with embedded models and techniques that effectively capture the processes governed by the network of molecular interactions including gene-gene, gene-protein, and protein-protein interactions, and can be customized for use in a variety of applications" [8]. It is set up as an open source community. As stated previously, Young [34] examined the use of BioSPICE modules JigCell and BioCharon to deterministically model systems of biochemical reactions. The use of four BioSPICE modules: Stochastica, Exact Stochastic Simulator, Gene Regulatory Analysis and Stochastic Simulation, and a University of North Carolina contribution, all of which

will be explained in chapter 3, are examined in this document to stochastically model systems of biochemical reactions.

III. Approaches

3.1 Overview

In this chapter, first the approaches and programs that will be used to solve stochastic differential equations are given, and then the approaches and programs that use Gillespie's exact method are presented. Note that instead of the values for the different species being considered as concentrations they will instead be considered as unitless quantities.

3.2 Stochastic Differential Equations

3.2.1 Euler-Maruyama. The explicit Euler-Maruyama method, described by equation (2.16), is used to examine the single differential equation (2.10) for the Hasty et al. protein regulation model, presented in chapter 2, when additive or multiplicative noise is involved. The stochastic differential equation is

$$\frac{dx(t)}{dt} \;=\; f(x(t)) \;+\; g(x(t))\xi(t) \tag{3.1}$$

where g(x(t)) = $\sqrt{\sigma^2}$ for the additive noise case, g(x(t)) = $x(t)\sqrt{\sigma^2}$ for the multiplicative noise case, and f(x(t)) is the right-hand side of the deterministic differential equation (2.10). Equation (3.1) will be examined with varying strengths of noise, where σ^2 is varied to change the noise strength.

The procedure used for the single differential equation is:

1. Groups of five hundred paths are calculated over a duration of twenty-five time units using different σ^2 values for each group. A Δt of 10^{-3} is used. Values are recorded at 0.1 time units.

2. The average X value for each group is then calculated at each point in time.

3. A final average for the X value of each group is calculated by taking an average of the last fifty time average values. The results with additive noise are presented in section 4.2.1.1. The results with multiplicative noise are presented in section 4.2.1.2.

The Wiener random variables are constructed in the following way:

$$W_0 \;=\; 0, \qquad\qquad W_{i+1} \;=\; W_i \;+\; (\sqrt{\Delta t})\,\eta_{i+1} \tag{3.2}$$

where $W_i = W(t_i)$ and $\eta_{i+1} \sim N(0,1)$. Note that with this construction, the Wiener random variables satisfy the properties for a Wiener process as described in section 2.3.1.

The implicit Euler-Maruyama method is used to examine the full system of differential equations, (2.6) through (2.9), for the protein regulation model when additive or multiplicative noise is involved. The implicit EM form is [24, pg 396]:

$$x_{i+1} \quad = \quad x_i \;+\; \Delta t f(x_{i+1}) \;+\; g(x_i)\Delta W_{i+1} \tag{3.3}$$

where $\Delta W_{i+1} = W_{i+1} - W_i$. Note that in equation (3.3) only the term representing the deterministic integral is evaluated at x_{i+1}. The g function is still evaluated at x_i because this is the Ito integral form.

The procedure used for the full system is the same as the single differential equation except that a total time of 50 time units is used because it takes longer for the full system when solved deterministically to come to its final steady state or at least very close to it. The average for each species is calculated. Whereas for the single differential equation g(x(t)) and $\xi(t)$ are both scalar values, for the full system g(x(t)) is a matrix and $\xi(t)$ is a vector. For this analysis the diagonal entries $g^{k,k}(x(t)) = \sqrt{\sigma^2}$ for the additive noise case and $g^{k,k}(x(t)) = \sqrt{\sigma^2}\,x^k(t)$ for the multiplicative noise case. All off diagonal entries equal zero. This is referred to

as diagonal noise because the kth species' concentration is directly disturbed only by the kth Wiener random variable [24, pg 348]. The results for the full system with additive noise are presented in section 4.2.1.1 and the results with multiplicative noise are presented in section 4.2.1.2.

3.2.1.1 Weak Convergence. The Euler-Maruyama method has order of weak convergence $\gamma = 1$ [16, pg 278] [24, pg 457]. The method described in Higham's paper [20] is used to examine the weak convergence of the method for equation 3.1 with additive noise g(x) = $\sqrt{\sigma^2}$ and multiplicative noise g(x) = $x\sqrt{\sigma^2}$. The log of both sides of equation 2.18 is taken to give the following:

$$log(\mid \mathbf{E}[X_L] - \mathbf{E}[X(T)] \mid) \quad \leq \quad log(C) \; + \; \gamma \, log(\Delta t) \tag{3.4}$$

Equation (3.4) provides an easy way to examine the order of weak convergence since it is in the slope intercept form of a line y = mx + b, where y = the left hand side, m = γ, and b = log(C). A line of slope γ should be parallel to the plot of the left hand side, obtained during numerical calculations, versus $log(\Delta t)$. Thus the order of convergence of the numerical calculations can be examined to see if it agrees with the theoretical order of convergence. The γ value obtained by the computations can be compared to the theoretical value by solving the linear system Ax = b with a least squares approach [20, pg 536] [9, pgs 315-333], where

$$A = \begin{pmatrix} 1 & log(\Delta t_1) \\ 1 & log(\Delta t_2) \\ 1 & log(\Delta t_3) \\ 1 & log(\Delta t_4) \\ 1 & log(\Delta t_5) \end{pmatrix} \qquad x = \begin{pmatrix} log(C) \\ \gamma \end{pmatrix} \qquad r = \begin{pmatrix} error(\Delta t_1) \\ error(\Delta t_2) \\ error(\Delta t_3) \\ error(\Delta t_4) \\ error(\Delta t_5) \end{pmatrix}$$

The entries in the r vector represent the errors obtained using the procedure, described momentarily, with respect to a path calculated using that particular Δt value.

The first procedure, performed with additive noise, is as follows [20, pg 537] [24, pg 458]:

1. Groups of fifty thousand paths using a particular σ^2 value are calculated using decreasing Δt values for each group over a duration of one time unit. Δt's of $2^{-9}, 2^{-8}, 2^{-7}, 2^{-6}$, and 2^{-5} are used.

2. The average values for the final time point are calculated for each of the differing Δt groups. The absolute error for each group is found by calculating the difference between the average final concentration and the deterministic value 1.2401. The deterministic value was calculated by solving the same system using the forward Euler method for one time unit.

3. The errors are plotted and compared to a reference line of slope one. The approximated γ value is also computed.

4. Steps 1 through 3 are performed ten times. If the calculated γ values are widely varying for each of the ten times, thus giving the appearance that the order of convergence is path dependent, then the σ^2 value is decreased. Steps 1 through 3 are performed ten more times. The process is stopped when consistent γ values are obtained for the ten times using the same σ^2 value. The results are presented in section 4.2.1.3.

Note that in his paper, Higham examined a linear model for which he had an analytical solution [20, pg 533]. Therefore he could find the expected true value directly. However, there does not exist an analytical solution for equation (3.1). The deterministic value is used in step 2 because the expected value of the SDE is assumed to result in a deterministic equation as follows:

$$\begin{aligned}\mathbf{E}[x(t)] &= \mathbf{E}[x(t_0)] + \mathbf{E}\left[\int_{t_0}^{t} f(x(\tau))d\tau\right] + \mathbf{E}\left[\sqrt{\sigma^2}\int_{t_0}^{t} dW(\tau)\right] \\ &= \mathbf{E}[x(t_0)] + \mathbf{E}\left[\int_{t_0}^{t} f(x(\tau))d\tau\right] + \sqrt{\sigma^2}\,\mathbf{E}[W(t) - W(t_0)] \qquad (3.5)\end{aligned}$$

Because the expected value of the Wiener random variables used is zero, equation (3.5) reduces to

$$\mathbf{E}[x(t)] = x(t_0) + \mathbf{E}\left[\int_{t_0}^{t} f(x(\tau))d\tau\right] \qquad (3.6)$$

The expected value of the integral involving the $f(x(\tau))$ term is assumed to be the deterministic result. This yields

$$\mathbf{E}[x(t)] = x(t_0) + \int_{t_0}^{t} f(x(\tau))d\tau \qquad (3.7)$$

Note that the procedure was not done for the case with multiplicative noise because there is not an analytic solution of the stochastic integral with x(t) as the integrand. The expected value of the integral in equation (3.6) may not be the deterministic value. For this reason a second procedure is also carried out. It involves using a path calculated with a very small Δt value to determine a close approximation to the true expected value. Other paths are created by sampling the path used to calculate the true expected value. This is the approach that Higham uses when he examines strong convergence for the Milstein method [20, pg 539].

The second procedure, performed with additive and multiplicative noise, is as follows:

1. Groups of five hundred paths using a specified σ^2 value are calculated using decreasing Δt values for each group over a duration of one time unit. Δt's of $2^{-11}, 2^{-7}, 2^{-6}, 2^{-5}$, and 2^{-4} are used. The paths using Δt's of $2^{-7}, 2^{-6}, 2^{-5}$, and 2^{-4} are samplings of the path with a Δt of 2^{-11}. First the Wiener random

variable increments for the $\Delta t = 2^{-11}$ path are created. The first increment for the $\Delta t = 2^{-7}$ path is created by summing the first 16 increments from the $\Delta t = 2^{-11}$ path, the second increment is created by summing the next 16 increments from the $\Delta t = 2^{-11}$ path, ... etc. The Wiener random variable increments are calculated the same way for the Δt paths of $2^{-6}, 2^{-5}$, and 2^{-4} using 32, 64, and 128 increments from the $\Delta t = 2^{-11}$ path respectively.

2. An expected final concentration is calculated for each of the Δt groups by averaging the final values of every path in the group. The absolute errors for the groups with Δt values of $2^{-7}, 2^{-6}, 2^{-5}$, and 2^{-4} are determined by calculating the differences between their average final concentration and the average final concentration for the $\Delta t = 2^{-11}$ path.

3. The errors are plotted and compared to a reference line of slope one. The approximated γ value is also computed.

4. Steps 1 through 3 are performed ten times. If the calculated γ values are widely varying for each of the ten times, thus giving the appearance that the order of convergence is path dependent, then the σ^2 value is decreased. Steps 1 through 3 are performed ten more times. The process is stopped when consistent γ values are obtained for the ten times using the same σ^2 value. The results are presented in section 4.2.1.3.

3.2.1.2 Strong Convergence. The Euler-Maruyama method has order of strong convergence $\gamma = 0.5$ [16, pgs 278] [24, pg 340]. The method in Higham's paper [20] is used to examine the strong convergence of the method for equation (3.1) with additive noise g(x) = $\sqrt{\sigma^2}$ and multiplicative noise g(x) = $x\sqrt{\sigma^2}$. The log of both sides of equation 2.17 is taken to give the following:

$$log(\mathbf{E}[\mid X_L - X(T) \mid]) \quad \leq \quad log(C) \; + \; \gamma \, log(\Delta t) \tag{3.8}$$

Similar to the method in section 3.2.1.1, equation (3.8) indicates that a reference line of slope γ can be used to examine the order of strong convergence of the numerical calculations as compared to the theoretical order of strong convergence. The approximated γ value can also be calculated as described in section 3.2.1.1.

There is not an analytical solution to equation (3.1), for this reason the same approach used in the second procedure in section 3.2.1.1 is carried out for strong convergence. A path with a very small Δt value is used to calculate an approximation to the true value.

The procedure, performed with additive and multiplicative noise, is as follows:

1. Groups of five hundred paths are calculated using decreasing Δt values for each group over a duration of one time unit. Δt's of $2^{-11}, 2^{-7}, 2^{-6}, 2^{-5}$, and 2^{-4} are used. The paths using Δt's of $2^{-7}, 2^{-6}, 2^{-5}$, and 2^{-4} are samplings of the path with a Δt of 2^{-11}. As with the second procedure for the weak convergence of the EM method, the Wiener random variable increments for the $\Delta t = 2^{-11}$ are created first. Each increment for the Δt paths of $2^{-7}, 2^{-6}, 2^{-5}$, and 2^{-4} is created by summing 16, 32, 64, and 128 increments of the $\Delta t = 2^{-11}$ path.

2. The absolute error is calculated for every path in the groups using Δt values of $2^{-7}, 2^{-6}, 2^{-5}$, and 2^{-4} by subtracting the final concentration from that obtained for the sampled $\Delta t = 2^{-11}$ path. The error for each of the four Δt groups is then calculated as the mean of its path errors.

3. The errors are plotted and compared to a reference line of slope one. The approximated γ value is also computed.

4. Steps 1 through 3 are performed ten times. If the calculated γ values are widely varying for each of the ten times, thus giving the appearance that the order of convergence is path dependent, then the σ^2 value is decreased. Steps 1 through 3 are performed ten more times. The process is stopped when consistent γ values are obtained for the ten times using the same σ^2 value. The results are presented in section 4.2.1.4.

3.2.2 Milstein's Method. The explicit Milstein's method [24, pg 345] is used to examine the single SDE (3.1) with the same multiplicative noise term, g(x) = $x\sqrt{\sigma^2}$, used for the EM method. By adding another term to the explicit Euler-Maruyama form, given in equation (2.16), the explicit Milstein method results. The extra term is obtained from the Ito-Taylor expansion [16, pgs 280-281] [24, pg 182]. The resulting explicit Milstein form is:

$$x_{i+1} \quad = \quad x_i \; + \; \Delta t f(x_i) \; + \; g(x_i)\Delta W_{i+1} \; + \; \frac{1}{2} g(x_i) \frac{\partial g(x_i)}{\partial x} ((\Delta W_{i+1})^2 - \Delta t) \quad (3.9)$$

where $\Delta W_{i+1} = W_{i+1} - W_i$. Note that if g were constant then this would reduce to the EM method. The procedure used solve the single equation is the same as that described for the EM method in section 3.2.1. The results are presented in section 4.2.2.

The implicit Milstein method is used to examine the full system of differential equations with diagonal multiplicative noise. The same form of diagonal noise defined for the implicit EM method, where $g^{k,k}(x) = x^k\sqrt{\sigma^2}$, is used. The resulting implicit Milstein form is [24, pgs 348,400]:

$$x^k_{i+1} \quad = \quad x^k_i \; + \; \Delta t f^k(\vec{x}_{i+1}) \; + \; g^{k,k}(\vec{x_i})\Delta W^k_{i+1} \; + \; \frac{1}{2} g^{k,k}(\vec{x_i}) \frac{\partial g^{k,k}(\vec{x_i})}{\partial x^k} ((\Delta W^k_{i+1})^2 - \Delta t) \quad (3.10)$$

where k represents the kth element of the vector and k,k represents a matrix entry. Note that the term involving the deterministic integral is implicit while the function g is evaluated at x_i. This is because the stochastic integral is being solved as an Ito integral. The procedure used for the full system is the same as that described for the EM method in section 3.2.1 to solve the single differential equation, except that a total time of 50 time units is used and the averages are calculated for each species. The results are presented in section 4.2.2.

3.2.2.1 Strong Convergence. The Milstein method has order of strong convergence $\gamma = 1$ [24, pg 345]. The second procedure stated in section 3.2.1.2 for examining the strong convergence of the EM method is used to examine the strong convergence of the Milstein method when solving equation (3.1) with multiplicative noise $g(x) = x\sqrt{\sigma^2}$. The results are presented in section 4.2.2.1.

3.2.3 Fokker-Planck. The Fokker-Planck equation is used to examine SDE (3.1) for the single differential equation for the protein regulation model. From equation (2.24), the Fokker-Planck equation has the following form:

$$\frac{\partial p(X,t)}{\partial t} = -\frac{\partial[f(X,t)p(X,t)]}{\partial X} + \frac{1}{2}\frac{\partial^2[g(X,t)^2 p(X,t)]}{\partial X^2} \tag{3.11}$$

where p(X,t) is the probability density function of X. The assumption is made that the system is at steady state. That is for large time $\frac{\partial p(X,t)}{\partial t} = 0$ and p(X,t) = p(X). Thus

$$\frac{d[f(X)p(X)]}{dX} = \frac{1}{2}\frac{d^2[g(X)^2 p(X)]}{dX^2} \tag{3.12}$$

Integrating equation (3.12) yields

$$\int_{-\infty}^{X} \frac{d[f(X)p(\tau)]}{d\tau}\, d\tau = \frac{1}{2}\int_{-\infty}^{X} \frac{d^2[g(\tau)^2 p(\tau)]}{d\tau^2}\, d\tau \tag{3.13}$$

Note that $p(\tau) \to 0$ as $\tau \to -\infty$. Therefore

$$f(X)p(X) = \frac{1}{2}\frac{d[g(X)^2 p(X)]}{dX} \tag{3.14}$$

Expanding the derivative term on the right hand side of equation (3.14) results in

$$2f(X)p(X) = 2g(X)\frac{d\,g(X)}{dX}p(X) + g(X)^2\frac{dp(X)}{dX} \tag{3.15}$$

If $g(X)^2 \neq 0$ in equation (3.15) then

$$\frac{dp(X)}{dX} \; + \; \frac{2}{g(X)^2}\left[-f(X) \; + \; g(X)\frac{d\,g(X)}{dX}\right]p(X) \quad = \quad 0 \tag{3.16}$$

Multiplying equation (3.16) by the integrating factor $\exp(2\phi(X))$, where

$$\frac{d\phi(X)}{dX} \quad = \quad \frac{1}{g(X)^2}\left[-f(X) \; + \; g(X)\frac{d\,g(X)}{dX}\right] \tag{3.17}$$

results in

$$\begin{aligned} 0 \quad &= \quad \exp\left(2\phi(X)\right)\frac{dp(X)}{dX} \; + \; 2\frac{d\phi(X)}{dX}p(X)\exp\left(2\phi(X)\right) \\ &= \quad \frac{d[p(X)\exp(2\phi(X))]}{dX} \end{aligned} \tag{3.18}$$

Integrating equation (3.18) and rearranging terms yields

$$p(X) \quad = \quad A\exp\left(-2\phi(X)\right) \tag{3.19}$$

This is the probability density function for the steady state value of X, where A is a normalization factor. That is

$$A \quad = \quad \frac{1}{\int_0^\infty \exp\left(-2\phi(X)\right)dX} \tag{3.20}$$

Using the definition of expected value results in the following equation [33, pgs 162-163]:

$$E[X_{ss}] \quad = \quad \int_0^\infty X\,A\exp\left(-2\phi(X)\right)dX \tag{3.21}$$

Note that the noise function is not visible in equation (3.23) because it is contained in $\phi(X)$ as described by equation (3.17). For the case of additive noise where g(X) $= \sqrt{\sigma^2}$, $\phi(X)$ has the following form:

$$\phi(X) \quad = \quad -\frac{1}{\sigma^2}\int_0^X f(\tau)\,d\tau \tag{3.22}$$

Equation (3.23) with $\phi(X)$ as defined in equation (3.22) is solved using the software package Mathcad for varying σ^2 values. The variances at steady state for varying the same σ^2 values are calculated using:

$$V[X_{ss}] \quad = \quad \int_0^\infty (X - E[X_{ss}])^2 A \exp\left(-2\phi(X)\right) dX \tag{3.23}$$

The results are presented in section 4.2.3.

For the case of multiplicative noise where g(X) $= X\sqrt{\sigma^2}$, $\phi(X)$ has the following form:

$$\phi(X) \quad = \quad \frac{1}{\sigma^2}\int_0^X \left[\frac{-f(\tau) \;+\; \sigma^2\tau}{\tau^2}\right] d\tau \tag{3.24}$$

It should be noted here that the multiplicative form only holds for $0 <$ X $< \infty$. Substituting the right hand side of equation 2.10 for the function f gives:

$$\phi(X) \;\; = \;\; \frac{1}{\sigma^2}\left[-\alpha\int_0^X \frac{1}{\beta+\tau^2}\,d\tau \;+\; (\gamma+\sigma^2)\int_0^X \frac{1}{\tau}\,d\tau \;-\; r\int_0^X \frac{1}{\tau^2}\,d\tau\right] \tag{3.25}$$

The singularities in the second and third integrals are very severe. These integrals are not Riemann or Lebesgue integrable. They are only integrable in the distributional sense [31, pgs 111-113].

Solving the multi-dimensional Fokker-Planck is a much more complicated task. In most cases analytic solutions will not be possible, numerical methods will be required. The four dimensional Fokker-Planck representation for the full system of differential equations, equations (2.6) through (2.9), results in a diffusion equation in four dimensional space. Using a finite difference scheme is impractical because of the large number of points at which the function would need to be evaluated. For example, if the four dimensional system was solved with each species restricted to vary on an interval [-5, 5] using increments of 0.1, this would result in 101 function evaluations in each spatial direction for a total of 101^4 function evaluations at each time step.

3.2.4 Gene Regulatory Analysis and Stochastic Simulation. GRASS is a BioSPICE module [5], in the BioSPICE 2.0 release, that solves chemical equations representative of gene regulation. It was created by Hasty, Collins, and McMillen. It uses a stochastic differential equation solver created by Elston and Adalsteinsson at the University of North Carolina. The version of the program, which is part of BioSPICE 2.0, is very specialized in the chemical reactions it solves and the user cannot enter just any general set of reactions and so a model other than the Hasty et al. protein model is examined. The specific gene system desired is entered in an input file format that does not require stoichiometric representations. However, stoichiometric representations of the reactions are given in an output file after running the simulation. A second output file contains the values of each species at each specified time unit. The output is in a form that can easily be loaded into Matlab. This was done so that means could be calculated. The program allows for the system to be solved deterministically or as a system of stochastic differential equations. An informative help manual is included with the program.

To examine GRASS a system of chemical reactions that model the genetic regulation of two proteins is considered. The descriptions used here can be found in the help manual included with the software. The system of reactions involves two

proteins and two DNA sites. Equations (3.26) through (3.35) describe the different species.

$$P_{1,1} \quad -- \quad \text{monomer form of protein 1} \tag{3.26}$$
$$P_{2,1} \quad -- \quad \text{monomer form of protein 2} \tag{3.27}$$
$$P_{1,2} \quad -- \quad \text{dimer form of protein 1} \tag{3.28}$$
$$P_{2,2} \quad -- \quad \text{dimer form of protein 2} \tag{3.29}$$
$$D_{1,0} \quad -- \quad \text{DNA site 1 with no protein bound} \tag{3.30}$$
$$D_{1,1} \quad -- \quad \text{DNA site 1 with protein 1 dimer bound} \tag{3.31}$$
$$D_{1,2} \quad -- \quad \text{DNA site 1 with protein 2 dimer bound} \tag{3.32}$$
$$D_{2,0} \quad -- \quad \text{DNA site 2 with no protein bound} \tag{3.33}$$
$$D_{2,1} \quad -- \quad \text{DNA site 2 with protein 1 dimer bound} \tag{3.34}$$
$$D_{2,2} \quad -- \quad \text{DNA site 2 with protein 2 dimer bound} \tag{3.35}$$

Equations (3.36) through (3.40) are the biochemical reactions that describe the production of the two proteins. Equations (3.36) and (3.39) describe the production of protein one and protein two respectively when there is nothing bound to the DNA sites. Equation (3.37) describes protein one acting as a repressor for itself when it is bound to the DNA site that codes for it. Note that it is a repressor because the reaction coefficient is less than the reaction coefficient for when nothing is bound to the DNA site in reaction (3.36). Thus during the same period of time, $D_{1,0}$ will produce more copies of protein one than $D_{1,1}$ will. Equation (3.38) describes protein two acting as an activator for protein one when it is bound to the DNA site that codes for protein one. It is an activator because the reaction coefficient is greater than that for reaction (3.36). Thus $D_{1,2}$ will produce more copies during the same period of time than $D_{1,0}$ will. In reaction (3.40) protein two neither acts as an activator or repressor for itself since the reaction coefficient is the same as for reaction (3.39). There is not an equation for when protein one is bound to the DNA site that produces protein two because protein one acts as a complete repressor and so no amount of protein two is made in this situation.

(Protein Production)

$$D_{1,0} \longrightarrow D_{1,0} + P_{1,1} \qquad k_1 = 5 \tag{3.36}$$

$$D_{1,1} \longrightarrow D_{1,1} + P_{1,1} \qquad k_2 = 1.25 \tag{3.37}$$

$$D_{1,2} \longrightarrow D_{1,2} + P_{1,1} \qquad k_3 = 6.25 \tag{3.38}$$

$$D_{2,0} \longrightarrow D_{2,0} + P_{2,1} \qquad k_4 = 4 \tag{3.39}$$

$$D_{2,2} \longrightarrow D_{2,2} + P_{2,1} \qquad k_5 = 4 \tag{3.40}$$

Equations (3.41) through (3.44) represent the degradation of the two proteins and their dimers, while (3.45) through (3.46) describe the creation of the dimers of protein one and two. Note that the empty parentheses signify that the proteins are degrading into a form that is not accounted for in the model.

(Protein Degradation)

$$P_{1,1} \longrightarrow ( ) \qquad k_6 = 0.7 \tag{3.41}$$

$$P_{1,2} \longrightarrow ( ) \qquad k_7 = 0.7 \tag{3.42}$$

$$P_{2,1} \longrightarrow ( ) \qquad k_8 = 0.5 \tag{3.43}$$

$$P_{2,2} \longrightarrow ( ) \qquad k_9 = 0.5 \tag{3.44}$$

(Protein Dimerization)

$$2P_{1,1} \longleftrightarrow P_{1,2} \qquad k_{10} = 0.5, \quad k_{-10} = 10 \tag{3.45}$$

$$2P_{2,1} \longleftrightarrow P_{2,2} \qquad k_{11} = 0.75, \quad k_{-11} = 10 \tag{3.46}$$

Equations (3.47) and (3.50) represent the binding of the protein one dimer to protein one's DNA site and the protein two dimer to protein two's DNA site. Equations (3.48) and (3.49) describe the binding of the protein one dimer and the protein two dimer to the other protein's DNA site.

(DNA-Protein Binding)

$$D_{1,0} + P_{1,2} \longleftrightarrow D_{1,1} \qquad k_{12} = 5, \quad k_{-12} = 10 \tag{3.47}$$

$$D_{1,0} + P_{2,2} \longleftrightarrow D_{1,2} \qquad k_{13} = 5, \quad k_{-13} = 10 \tag{3.48}$$

$$D_{2,0} + P_{1,2} \longleftrightarrow D_{2,1} \qquad k_{14} = 5, \qquad k_{-14} = 10 \tag{3.49}$$

$$D_{2,0} + P_{2,2} \longleftrightarrow D_{2,2} \qquad k_{15} = 5, \qquad k_{-15} = 10 \tag{3.50}$$

Using the rate equation approach, the following system of differential equations is derived:

$$\frac{d[P_{1,1}]}{dt} = k_1[D_{1,0}] + k_2[D_{1,1}] + k_3[D_{1,2}] - k_6[P_{1,1}] - 2k_{10}[P_{1,1}]^2 + 2k_{-10}[P_{1,2}] \tag{3.51}$$

$$\frac{d[P_{2,1}]}{dt} = k_4[D_{2,0}] + k_5[D_{2,2}] - k_8[P_{2,1}] - 2k_{11}[P_{2,1}]^2 + 2k_{-11}[P_{2,2}] \tag{3.52}$$

$$\frac{d[P_{1,2}]}{dt} = -k_7[P_{1,2}] + k_{10}[P_{1,1}]^2 - k_{-10}[P_{1,2}] - k_{12}[P_{1,2}][D_{1,0}] + k_{-12}[D_{1,1}] - k_{14}[P_{1,2}][D_{2,0}] + k_{-14}[D_{2,1}] \tag{3.53}$$

$$\frac{d[P_{2,2}]}{dt} = -k_9[P_{2,2}] + k_{11}[P_{2,1}]^2 - k_{-11}[P_{2,2}] - k_{13}[P_{2,2}][D_{1,0}] + k_{-13}[D_{1,2}] - k_{15}[P_{2,2}][D_{2,0}] + k_{-15}[D_{2,2}] \tag{3.54}$$

$$\frac{d[D_{1,0}]}{dt} = -k_{12}[P_{1,2}][D_{1,0}] + k_{-12}[D_{1,1}] - k_{13}[P_{2,2}][D_{1,0}] + k_{-13}[D_{1,2}] \tag{3.55}$$

$$\frac{d[D_{1,1}]}{dt} = k_{12}[P_{1,2}][D_{1,0}] - k_{-12}[D_{1,1}] \tag{3.56}$$

$$\frac{d[D_{1,2}]}{dt} = k_{13}[P_{2,2}][D_{1,0}] - k_{-13}[D_{1,2}] \tag{3.57}$$

$$\frac{d[D_{2,0}]}{dt} = -k_{14}[P_{1,2}][D_{2,0}] + k_{-14}[D_{2,1}] - k_{15}[P_{2,2}][D_{2,0}] + k_{-15}[D_{2,2}] \tag{3.58}$$

$$\frac{d[D_{2,1}]}{dt} = k_{14}[P_{1,2}][D_{2,0}] - k_{-14}[D_{2,1}] \tag{3.59}$$

$$\frac{d[D_{2,2}]}{dt} = k_{15}[P_{2,2}][D_{2,0}] - k_{-15}[D_{2,2}] \tag{3.60}$$

The procedure is as follows:

1. Twenty paths are calculated in GRASS, Stochastica(Stochastica is described in section 3.3.1), and using the implicit Euler-Maruyama method with additive diagonal noise, where $g_{i,i}(x) = \sqrt{\sigma^2}$. A Δt of 10^{-4} with $\sigma^2 = 5$ is used for the EM calculations. The values are recorded every 0.01 time steps for each approach. An example of the input file used for GRASS is given in appendix A. Only the seed value for the random number generator was changed for each calculation. The deterministic solution is also calculated using Grass, the Matlab ode23s function, and the backward Euler method. The Matlab ode23s function is a stiff solver for differential equations. A Δt of 10^{-4} is used for the backward Euler method

2. The average value for each species is then calculated at each point in time.

3. A final average for the each species' value is calculated by taking an average of the last two hundred and fifty time average values. The results are reported in section 4.2.4.

3.3 Gillespie's Exact Method

3.3.1 Stochastica. Stochastica is a BioSPICE module, in the BioSPICE 2.0 release, that implements Gillespie's exact method. It is a contribution from Beckwith at UCLA. A very helpful user manual is supplied with the program [4]. It provides a nice summary of the algorithm used and discusses some example input files included with the program. The input files are easily created. An example of an input file is given in the Appendix A.1. The user specifies the chemical reactions in stoichiometric form and declares the rate coefficients. Then the user declares the duration of the reaction, the time intervals at which the values for species are to be recorded, and the total number of paths that are to be calculated. The recorded values for each of the species is written to different text files. The recorded values in the output files are easily loaded into Matlab where the averages are calculated.

The procedure is as follows:

1. Groups of one thousand paths are calculated using Stochastica and a Matlab implementation of the direct method, the algorithm presented in section 2.3.3,

for a duration of one hundred time units for varying parameter and initial species values. The values are recorded at 0.1 time units. The deterministic solution is also calculated using Matlab's ode23s function.

2. For each group the average value for each of the species is then calculated at each time interval.

3. A final average for each species is calculated by taking an average of the last two hundred time average values. The results are presented in section 4.3.1

3.3.2 Exact Stochastic Simulator. Exact Stochastic Simulator (ESS) is a BioSPICE module, in the BioSPICE 3.0 release, that also implements Gillespie's exact method. It was written by McCollum and Lancaster at the University of Tennesse-Knoxville [10]. ESS is only one part of the package. Two other programs, BioSpreadsheet and BioSmokey, are included. BioSpreadsheet is a GUI that is used to create an input file for the ESS program. BioSmokey is a program that performs different functions on the ESS output file. Example functions are: extracting the values at all times for a particular species, finding the mean of a species for a path, and finding the variance of a species for a path. A nice feature is that the user enters the seed value for the random number generator when doing simulations so that results for a particular path can be replicated. A major drawback of the program is that it does not allow the user to carry out multiple simulations at one time. The user needs to enter a seed value for each simulation. This made the process of calculating multiple paths tedious and time consuming to carry out. The output is not in a form that is easily loaded into Matlab where the means are calculated. The same procedure, described in section 3.3.1, used for the Stochastica calculations is performed for ESS with the exception that 20 paths instead of 1,000 are used. The results are presented in section 4.3.2.

3.3.3 τ-Leaping. Gillespie has created a variation, called τ-leaping, of his exact method, described in section 2.3.3, which reduces the number of reactions that

need to be computed during the simulation [15]. The idea is that if in a very small amount of time a number of reactions are going to occur and the resulting change in the values of the different species does not significantly change the probability that the reaction will occur in the next very small time step, then leaping over a small period of time can speed up the simulation while still maintaining the true nature of the system. There are two different implementations of the τ-leaping method: plain τ-leaping and estimated-midpoint τ-leaping.

The plain τ-leap algorithm is as follows [15, pgs 1720-1721]:

1. Set reaction coefficients and initial numbers of molecules for all species. Specify a stopping time and the times to record the molecule values for all species.

2. Using the state vector $\vec{x}^T = (x_1, x_2, ..., x_N)$, calculate $h(\vec{x})_i$, the number of combinations in which the required reactants can interact in order for the ith reaction to occur, using the following formula [33, pg 44]:

$$h(\vec{x})_i = \frac{x_1!}{r_{i,1}!(x_1 - r_{i,1})!}\frac{x_2!}{r_{i,2}!(x_2 - r_{i,2})!} ... \frac{x_N!}{r_{i,N}!(x_N - r_{i,N})!} \qquad i = 1, 2, 3, ...M \tag{3.61}$$

   where M is the total number of reactions, N is the total number of species, x_j indicates the number of molecules of the jth species, and $r_{i,j}$ indicates the stoichiometric coefficient for the jth species in the ith reaction. Note that the stoichiometric coefficient for a species which does not appear in the ith reaction is equal to zero. Gillespie provides some examples for commonly encountered reactions [13, pgs 405, 413].

3. Calculate

$$a(\vec{x})_i = c_i\, h(\vec{x})_i \qquad i = 1, 2, 3, ...M \tag{3.62}$$

   where c_i is the reaction rate for the ith reaction, $h(\vec{x})_i$ is described in step 2, and $a(\vec{x})_i$ is the propensity function for the ith reaction. That is when $a(\vec{x})_i$ is multiplied by a suitably small change in time it gives the probability that the ith reaction will occur in the next time step [15, pg 1717]. Equation (3.62) is the same as Gillespie's equation (25) [13, pg 415].

4. A time leap value τ is calculated with:

$$\tau \quad = \quad \min_{i=1,\ldots M} \frac{\epsilon \; a(\vec{x})_0}{\mid \sum_{j=1}^{N} \xi(\vec{x})_j \; b(\vec{x})_{i,j} \mid} \tag{3.63}$$

where $(0 < \epsilon < 1)$, and

$$a(\vec{x})_0 \quad = \quad \sum_{i=1}^{M} a(\vec{x})_i \tag{3.64}$$

$$\xi(\vec{x}) \quad = \quad \sum_{i=1}^{M} a(\vec{x})_i \, \vec{v_i} \tag{3.65}$$

$$b(\vec{x})_{i,j} \quad = \quad \frac{\partial a(\vec{x})_i}{\partial x_j} \tag{3.66}$$

Note that while $a(\vec{x})_i$ is a scalar, $\vec{v_i}$ is a vector. The jth row entry of $\vec{v_i}$ indicates the change in the jth species that occurs when the ith reaction takes place. Equation (3.63) is the same as Gillespie's equation (26a) [15, pg 1721].

5. Use the exact method and skip to step 8 if

$$\tau \quad \leq \quad \frac{2}{a(\vec{x})_0} \tag{3.67}$$

Note that using 2 here states that if τ is less than 2 times the expected time until the next reaction occurs, don't do a leap. Other values besides 2 could be used [15, pg 1721].

6. The number of times that the ith reaction will occur in the next small time step τ is calculated by generating a Poisson random number

$$k(\tau, \vec{x}, t)_i \quad = \quad P(a(\vec{x})_i, \tau) \tag{3.68}$$

where $a(\vec{x})_i \, \tau$ is the mean of the Poisson distribution being sampled. Equation (3.68) is the same as Gillespie's equation (16) [15, pg 1719].

7. Calculate the change of values for the species after the time step τ

$$\vec{\lambda} \quad = \quad \sum_{i=1}^{M} k(\tau, \vec{x}, t)_i \, \vec{v_i} \tag{3.69}$$

Equation (3.69) is the same as Gillespie's equation (18) [15, pg 1719].

8. Update the values

$$\vec{x} \quad = \quad \vec{x} + \vec{\lambda} \tag{3.70}$$

9. If the time is greater than or equal to the stopping time quit, otherwise, return to step 2.

Steps 1, 2, and 3 are the exact same as for the exact method algorithm given in section 2.3.3. The ϵ term in step 4 is used to "bound the expected changes in the propensity functions in time τ" [15, pg 1720] calculated in step 3.

The estimated-midpoint algorithm is obtained by substituting the following four steps for step 5 in the plain τ-leap algorithm [15, pgs 1721-1722]:

5a The expected change in the values $\overline{\vec{\lambda}}$ during the time step τ is calculated:

$$\overline{\vec{\lambda}} \; = \; \tau \sum_{i}^{M} a(\vec{x})_i \vec{v}_i \tag{3.71}$$

Equation (3.71) is the same as Gillespie's equation (21) [15, pg 1720].

5b The expected midpoint is now calculated

$$\overline{\vec{x}} \quad = \quad \vec{x} + \frac{\overline{\vec{\lambda}}}{2} \tag{3.72}$$

5c The $a(\vec{x})_i$'s, the propensity functions for each reaction, are recalculated using the expected midpoint

$$a(\overline{\vec{x}})_i = c_i \, h(\overline{\vec{x}})_i \qquad i = 1, 2, 3, ...M \tag{3.73}$$

5d The number of times that the ith reaction will occur in the next small time step τ is calculated by generating a Poisson random number

$$k(\tau, \overline{\vec{x}}, t)_i \quad = \quad P(a(\overline{\vec{x}})_i, \tau) \tag{3.74}$$

where $a(\overline{\vec{x}})_i \, \tau$ is the mean of the Poisson distribution being sampled.

To better illustrate the use of the plain τ-leap algorithm, some of the equations resulting for a simple example are shown using the following reactions with forward and reverse reaction coefficients of c_f and c_r:

$$A \; + \; B \; \rightleftarrows \; 2C$$

The forward reaction is reaction 1 and the backward reaction is reaction 2. The total possible combinations for the interactions of the molecules that lead to the forward and reverse reactions are given by h_1 and h_2 respectively:

$$h_1 \; = \; \frac{[A]!}{1!\,([A]-1)!}\frac{[B]!}{1!\,([B]-1)!} \; = \; [A]\,[B] \qquad\qquad h_2 \; = \; \frac{[C]!}{2!\,([C]-2)!} \; = \; \frac{[C]\,([C]-1)}{2}$$

The propensity functions a_1 and a_2 are:

$$a_1 \; = \; c_f h_1 \; = \; c_f [A][B], \qquad\qquad a_2 \; = \; c_r h_2 \; = \; c_r \frac{[C]([C]-1)}{2}$$

The vectors v_1 and v_2 that indicate the changes in the different species values for the reactions are:

$$v_1 \; = \; \begin{pmatrix} -1 \\ -1 \\ 2 \end{pmatrix} \qquad\qquad v_2 \; = \; \begin{pmatrix} 1 \\ 1 \\ -2 \end{pmatrix}$$

The b matrix used to calculate the time step τ has the form:

$$b \; = \; \begin{pmatrix} c_f\,[B] & c_f\,[A] & 0 \\ 0 & 0 & c_r\,([C]-\frac{1}{2}) \end{pmatrix}$$

The procedure is as follows:

1. Groups of one thousand paths are calculated using Matlab implementations of the direct method, the algorithm presented in section 2.3.3, and the plain and estimated-midpoint τ-leaping methods for a duration of one hundred time units for varying parameter and initial species values. The values are recorded at 0.1 time units. Two sets of the groups are calculated for the τ-leaping methods using ϵ values of 0.05 and 0.15. For each of the paths calculated using the direct method, the number of reactions simulated is recorded, while for paths calculated using the τ-leaping methods the number of exact reactions simulated and the number of leaps performed is recorded.

2. For each group the average value for each of the species at each time interval, the average number of exact reactions, and the average number of leaps is calculated.

3. A final average for each species is calculated by taking an average of the last two hundred time average values. The results are presented in section 4.3.3

3.4 UNC Module

The UNC module is a program in the BioSPICE 3.0 release [5] that solves chemical equations representative of genetic regulation. It is a contribution from Elston and Adalsteinsson at the University of North Carolina. The documentation that is included with the software states that the program was created with the intention of running on an Apple computer but that it could also be run in a linux environment. Instructions for running make commands that create executable files are given. This included creating three input files: one that describes the reaction coefficients, one for the initial values, and one that declares the other parameters needed, such as duration and time intervals to record values. Problems were encountered when performing the make commands. After examining the source code that is included in the package, it was discovered that a simple change in the Information.cpp file, changing the number that the length of an input array was compared to, would allow the program to compile and run the example correctly. Correspondence with the author indicated that it was only necessary to make changes to the

Rates.cpp and Matrices.cpp files to get another system of chemical equations to be solved. A close examination of the code seems to indicate that the code has been written specifically to solve the example problem that is included. For example, the SolveImplicit.cpp file has the Jacobian matrix for the example problem hard coded in the program.

The idea behind the UNC program is to give the user choices of approaches to use to solve the system of chemical equations. There are three different approaches given for solving the chemical equations: a fully discrete method, a fully continuous method, and a hybrid method that incorporates both discrete and continuous solving. There is no explanation of any of the approaches in the documentation.

To examine the UNC module the system of chemical reactions that is included with the program are considered. Equations (3.75) through (3.83) describe the different species involved.

$$A \quad -- \quad \text{protein} \tag{3.75}$$

$$R \quad -- \quad \text{protein} \tag{3.76}$$

$$mRNA.A \quad -- \quad \text{messenger RNA that codes for protein A} \tag{3.77}$$

$$mRNA.R \quad -- \quad \text{messenger RNA that codes for protein R} \tag{3.78}$$

$$AR \quad -- \quad \text{molecule of protein A bound to protein R} \tag{3.79}$$

$$DA \quad -- \quad \text{DNA site that codes for protein A} \tag{3.80}$$

$$DR \quad -- \quad \text{DNA site that codes for protein R} \tag{3.81}$$

$$DAA \quad -- \quad \text{DNA site that codes for protein A, bound to a protein A} \tag{3.82}$$

$$DRA \quad -- \quad \text{DNA site that codes for protein R, bound to a protein A} \tag{3.83}$$

Equations (3.84) and (3.85) describe transcription at the two DNA sites resulting in the creation of the messenger RNA's. Equations (3.86) and (3.87) describe transcription when a protein A is bound to the DNA site. It can be seen that protein A acts as an activator for protein A and protein R by observing that the reaction rates when protein A is bound to the DNA sites are increased.

$$DA \longrightarrow DA + mRNA.A \qquad k_1 = 50 \tag{3.84}$$
$$DR \longrightarrow DR + mRNA.R \qquad k_2 = 0.01 \tag{3.85}$$
$$DAA \longrightarrow DAA + mRNA.A \qquad k_3 = 500 \tag{3.86}$$
$$DRA \longrightarrow DRA + mRNA.R \qquad k_4 = 50 \tag{3.87}$$

Equations (3.88) and (3.89) represent the binding of protein A to the DNA sites.

$$DA + A \longleftrightarrow DAA \qquad k_5 = 1, \qquad k_{-5} = 50 \tag{3.88}$$
$$DR + A \longleftrightarrow DRA \qquad k_6 = 1, \qquad k_{-6} = 100 \tag{3.89}$$

Equations (3.90) and (3.91) describe translation for protein A and protein R respectively.

$$mRNA.A \longrightarrow mRNA.A + A \qquad k_7 = 500 \tag{3.90}$$
$$mRNA.R \longrightarrow mRNA.R + R \qquad k_8 = 100 \tag{3.91}$$

Equations (3.92) and (3.93) represent the degradation of proteins A and R while equations (3.94) and (3.95) represent the degradation of the messenger RNA's. Note that the empty parentheses signify that the proteins and messenger RNA's are degrading into a form that is not accounted for in the model.

$$A \longrightarrow ( ) \qquad k_9 = 1 \tag{3.92}$$
$$R \longrightarrow ( ) \qquad k_{10} = 0.2 \tag{3.93}$$
$$mRNA.A \longrightarrow ( ) \qquad k_{11} = 10 \tag{3.94}$$
$$mRNA.R \longrightarrow ( ) \qquad k_{12} = 0.5 \tag{3.95}$$

Equation (3.96) describes the coupling of proteins A and R. Equation (3.97) describes the destruction of protein A by protein R.

$$A + R \longrightarrow AR \qquad k_{13} = 20 \tag{3.96}$$
$$AR \longrightarrow R \qquad k_{14} = 10 \tag{3.97}$$

Using the rate equation approach, the following system of differential equations is derived:

$$\frac{d[A]}{dt} = -k_5[DA][A] + k_{-5}[DAA] - k_6[DR][A] + k_{-6}[DRA] + k_7[mRNA.A] - k_9[A] - k_{13}[A][R] \quad (3.98)$$

$$\frac{d[AR]}{dt} = k_{13}[A][R] - k_{14}[AR] \quad (3.99)$$

$$\frac{d[DA]}{dt} = -k_5[DA][A] + k_{-5}[DAA] \quad (3.100)$$

$$\frac{d[DAA]}{dt} = k_5[DA][A] - k_{-5}[DAA] \quad (3.101)$$

$$\frac{d[DR]}{dt} = -k_6[DR][A] + k_{-6}[DRA] \quad (3.102)$$

$$\frac{d[DRA]}{dt} = k_6[DR][A] - k_{-6}[DRA] \quad (3.103)$$

$$\frac{d[R]}{dt} = k_9[mRNA.R] - k_{10}[R] - k_{13}[A][R] + k_{14}[AR] \quad (3.104)$$

$$\frac{d[mRNA.A]}{dt} = k_1[DA] - k_{11}[mRNA.A] + k_3[DAA] \quad (3.105)$$

$$\frac{d[mRNA.R]}{dt} = k_2[DR] - k_{12}[mRNA.R] + k_4[DRA] \quad (3.106)$$

The procedure is as follows:

1. Three groups of twenty paths are calculated in UNC, with each one using one of the different approaches: fully continuous, fully discrete, and the hybrid method with a Δt of 0.001. Species values are recorded at 0.01 time intervals. Twenty paths are also calculated in Stochastica and using the implicit Euler-Maruyama method with additive diagonal noise, where $g_{i,i}(x) = \sqrt{\sigma^2}$. A Δt of 1/75000 with $\sigma^2 = 5$ is used for the EM calculations. The values are recorded every 0.01 time steps for each approach. The deterministic solution is calculated using Matlab's ode23s function and the backward Euler method. A Δt of 1/75000 is used for the backward Euler method. A total time duration

of 10 time units is used.

2. For each group the average value for each species is then calculated at each point in time.

3. A final average for the each species' value is calculated by taking an average of the last twenty time average values. The results are presented in section 4.4.

It was necessary to change the default seed values for the random number generator in the Mersanne.cpp file to get calculate a different path each time the program was run. Otherwise, the same results were obtained when running the program multiple times. Creating a seed input file as described by the directions did not seem to produce different results. During the hybrid method, reactions (3.88) and (3.89) are solved using the discrete approach while the others are solved using the continuous approach. No reason is given as to the choice of reactions to solve in a discrete or continuous manner.

IV. Results

4.1 Overview

The results using the different techniques and software programs described in Chapters 2 and 3 are reported and discussed in this chapter. It should be understood that while a deterministic solution can come to a steady state, the stochastic solutions can only come to a somewhat steady state. The term somewhat steady state is used because the system is constantly undergoing random change even though the perturbations might be very small. The Hasty et al. protein regulation model was examined using the Euler-Maruyama method, the Milstein method, the Stochastica and ESS programs, and the τ-leaping method. The different parameter values used for these approaches are described in their respective sections. Using the parameter values, the critical values for the X species were calculated by setting the right hand side of equation (2.10) equal to zero and then solving for the roots of the resulting third degree polynomial. After the critical values for the X species were calculated, the following equations were used to obtain the critical values for the other species for the full system [7, pg 3-21]:

$$[X_2] = \frac{k_1}{k_{-1}}[X]^2 \tag{4.1}$$

$$[D] = (1 + \frac{k_2}{k_{-2}}X_2)^{-1}dT \tag{4.2}$$

$$[DX_2] = \frac{k_2}{k_{-2}}[D][X_2] \tag{4.3}$$

Note that as was done in chapter 3, instead of the values for the different species being considered as concentrations they will instead be considered as unitless quantities.

Table 4.1: Model Parameter Values Used for the Euler-Maruyama and Milstein Methods

n	p	r	k_1	k_{-1}	k_2	k_{-2}	k_3	k_4	d_T
2	2	0.4	10	5	5	10	2.5	10	2

Table 4.2: Resulting Critical Values of the Model Used for the Euler-Maruyama and Milstein Methods

	X	X_2	D	DX_2
Stable	1.2873	3.3145	0.7527	1.2473
Unstable	0.7088	1.0049	1.3312	0.6688
Stable	0.0438	0.0038	1.9962	0.0038

4.2 Stochastic Differential Equations

The parameter values used for the Hasty et al. protein regulation model [17] when using the Euler-Maruyama and Milstein methods are given in Table 4.1 and the resulting critical values are given in Table 4.2. The different initial conditions used for the model when using the Euler-Maruyama and Milstein methods are given in Table 4.3. Note that throughout chapter 4, the critical values will be referred to as lower, middle, and upper. These relations are in reference to the X species values. The lower critical value is the one which corresponds to the lowest X value. The middle and upper critical values are similarly defined. The values for the X_2, D, and DX_2 that correspond to the lower critical value for the X species are defined to be in their lower critical values. The same can be said for the middle and upper critical values. The first set of initial conditions was chosen because the X value was halfway between the lower critical value and the middle critical value. For the second and third sets of initial conditions, the X values are small perturbations away from the middle critical value towards the lower and upper critical values respectively. The fourth set of initial conditions was chosen because the X value was about halfway between the middle critical value and the upper critical value. The σ^2 values used to vary the strength of the noise are given in Table 4.4.

Table 4.3: Model Initial Conditions Used for the Euler-Maruyama and Milstein Methods

Set	X	X_2	D	DX_2
1	0.37	0.50	1.6	0.4
2	0.70	0.80	1.4	0.6
3	0.71	1.10	1.0	1.0
4	0.95	2.15	1.1	0.9

Table 4.4: σ^2 Values used for the Euler-Maruyama and Milstein Methods

	Trial												
	1	2	3	4	5	6	7	8	9	10	11	12	13
σ^2	0.00001	0.00005	0.0001	0.0005	0.001	0.005	0.01	0.05	0.1	0.25	0.5	0.75	1

4.2.1 Euler-Maruyama Method. Because equation (3.3) is implicit and the $f(\vec{x}_{i+1})$ term is nonlinear, it was necessary to use a Taylor series to linearize the problem. The $f(\vec{x}_{i+1})$ term is expanded:

$$\begin{aligned} f(\vec{x}_{i+1}) &= f(\vec{x}_i + \triangle x) \\ &\approx f(\vec{x}_i) + J(\vec{x}_i)(\vec{x}_{i+1} - \vec{x}_i) \end{aligned} \tag{4.4}$$

where J is the Jacobian matrix. That is

$$J_{j,\,k}(\vec{x}_i) = \frac{\partial f_j(\vec{x}_i)}{\partial x_k} \tag{4.5}$$

The expanded $f(\vec{x}_{i+1})$ term is then substituted into equation (3.3).

$$\vec{x}_{i+1} = \vec{x}_i + \triangle t f(\vec{x}_i) + \triangle t J(\vec{x}_i)\vec{x}_{i+1} - \triangle t J(\vec{x}_i)\vec{x}_i + g(\vec{x}_i)\overrightarrow{\triangle W}_{i+1} \tag{4.6}$$

All terms involving $\vec{x}_{i+1}$ are isolated on the left hand side.

$$(I - \triangle tJ(\vec{x}_i))\vec{x}_{i+1} \quad = \quad \vec{x}_i + \triangle tf(\vec{x}_i) - \triangle tJ(\vec{x}_i)\vec{x}_i + g(\vec{x}_i)\overrightarrow{\triangle W}_{i+1} \qquad (4.7)$$

Multiplying both sides of equation (4.7) by the inverse of $(I - \triangle tJ(\vec{x}_i))$ yields the form used for calculations:

$$\vec{x}_{i+1} \quad = \quad (I - \triangle tJ(\vec{x}_i))^{-1}(\vec{x}_i + \triangle tf(\vec{x}_i) - \triangle tJ(\vec{x}_i)\vec{x}_i + g(\vec{x}_i)\overrightarrow{\triangle W}_{i+1}) \quad (4.8)$$

where I is the identity matrix, f is the right-hand side of the deterministic differential equation, J is the Jacobian matrix, g is the diagonal noise matrix, and the jth component of $\overrightarrow{\triangle W}_{i+1}$ equals $(\sqrt{\Delta t})\eta$, with $\eta \sim N(0,1)$.

4.2.1.1 Additive Noise. The mean final values resulting from each σ^2 value for the single differential equation with additive noise, as per the procedure described in section 3.2.1, are shown graphically in Figures 4.1 and 4.2. The percentages of path values at the final time that were near the upper or lower critical values are graphed for the different trials in Figures 4.3 and 4.4. Any values less than the middle critical value were defined to be near the lower critical value and all others were defined to be near the upper critical value.

For trials 1 through 7, which correspond to the different σ^2 values given in Table 4.4, using the first initial condition, shown in Figure 4.3, not many of the values switched across the middle critical value to end up near the upper critical value. With increased noise, as shown by trials 10 through 13, the majority of final path values were near the upper critical value.

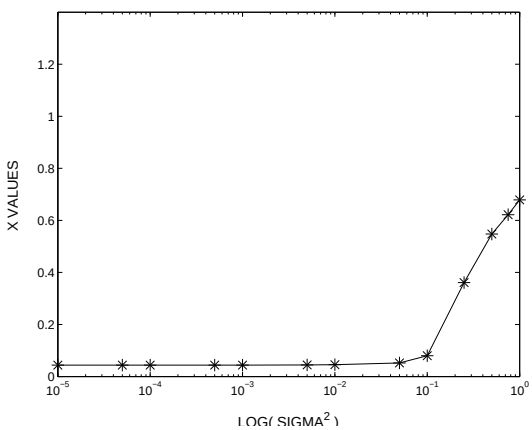
The results for the second initial condition, given in Figure 4.1, and the results for the third initial condition, given in Figure 4.2, show similar mean final value

tendencies for trials 8 through 13. The majority of the final path values are near the upper critical value for these trials, as shown in Figures 4.3 and 4.4. As the σ^2 values decrease, the switching behavior decreases for the second initial condition, and for trials 1 and 2 all final path values end near the lower critical value. However, the final path values for the third initial condition still shows a significant amount of switching behavior as some of the final path values are near the lower critical value. It should be noted though, that the third initial condition is a much smaller perturbation away from the middle critical value than the second initial condition.

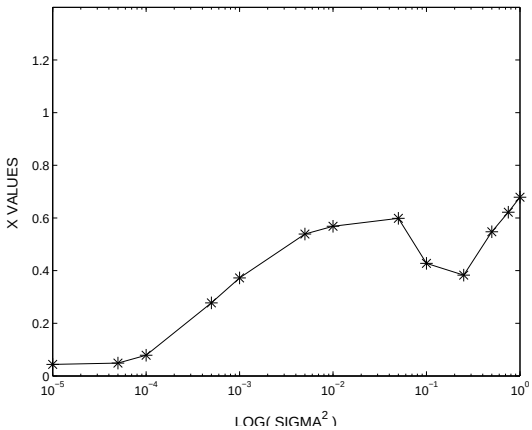
For the fourth initial condition, given in Figure 4.2, the mean values are close to the upper critical value for trials 1 through 7. No switching occurred, all of the final path values were near the upper critical value, Figure 4.4. For the larger σ^2 values, the majority of final path values ended near the upper critical value. Trials 10 through 13 for all initial conditions show similar mean final values and percentages.

As σ^2 tended to zero the final average values tended to the deterministic results, obtained using the forward Euler method, for all the initial conditions. As σ^2 was increased this caused the values that began near the lower critical value to exhibit an increased tendency to switch across the middle critical value and end near the upper critical value. For the largest σ^2 values, trials 10 through 13, the majority of paths with initial values near the lower critical value switched to the upper critical value, as shown in Figures 4.3 and 4.4. In contrast, for no σ^2 values did paths with initial values near the upper critical value ever have a majority of final path values near the lower critical value.

When performing the calculations for the full system with σ^2 values above 0.01, warnings were sometimes given by Matlab that indicated the matrix that is inverted during a time step was nearly singular. The added random values were causing values to become either very large or very small. Even after the paths for which this occurred were thrown out, the final calculated values were unrealistic since there were extremely large values and negative values. For this reason trials



(a) mean final X values initial condition 1: x = 0.37, for 25 time units, 500 paths used, critical values: 0.0438, 0.7088, 1.2873



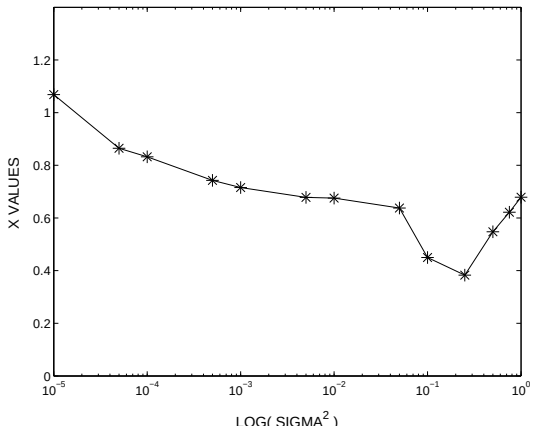
(b) mean final X values initial condition 2: x = 0.7, for 25 time units, 500 paths used, critical values: 0.0438, 0.7088, 1.2873

Figure 4.1: Euler-Maruyama Mean Values for Single SDE, Additive Noise, Initial Conditions 1 and 2

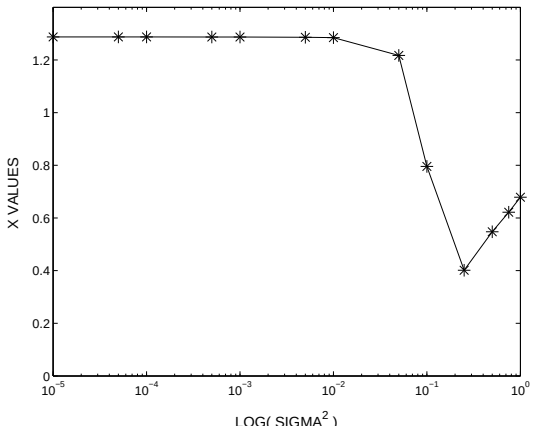
8 through 13 are not reported for the full system, with the exception of trial 8 for the fourth set of initial conditions. The mean final values of all species for the first and second set of initial conditions are given in Table 4.5 and in Table 4.6 for the third and fourth set of initial conditions. The column under the "% upper" heading indicates the percentage of path final time values for X that were greater than or equal to the middle critical value. The procedure followed is the same as for the single differential equation presented in section 3.2.1, except that the calculations are carried out for 50 time units and averages are calculated for each species.

It only took a few minutes to calculate all the paths for one set of initial conditions for the single equation. However, the calculations for the full system took multiple hours. The amount of time that elapsed to compute a single path for the full system, using the first set of initial conditions with a σ^2 value of 0.01, was calculated using the clock and etime functions in Matlab. The elapsed time for a single path was found to be 15.1250 seconds on a 2.4 gigahertz system.

The results for the full system for the first set of initial conditions, Table 4.5, for trials 6 and 7 show increased switching behavior to the upper critical value when



(a) mean final X values initial condition 3: x = 0.71, for 25 time units, 500 paths used, critical values: 0.0438, 0.7088, 1.2873



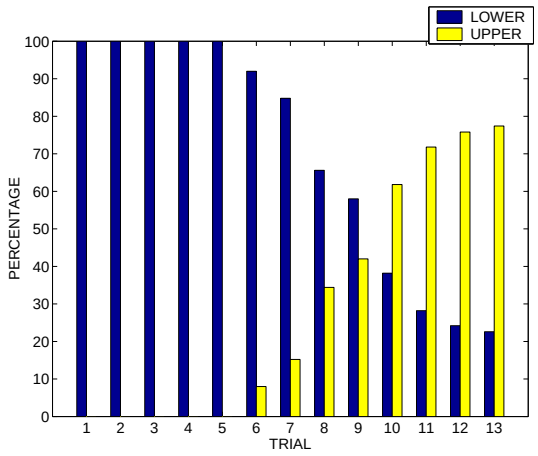
(b) mean final X values initial condition 4: x = 0.95, for 25 time units, 500 paths used, critical values: 0.0438, 0.7088, 1.2873

Figure 4.2: Euler-Maruyama Mean Values for Single SDE, Additive Noise, Initial Conditions 3 and 4

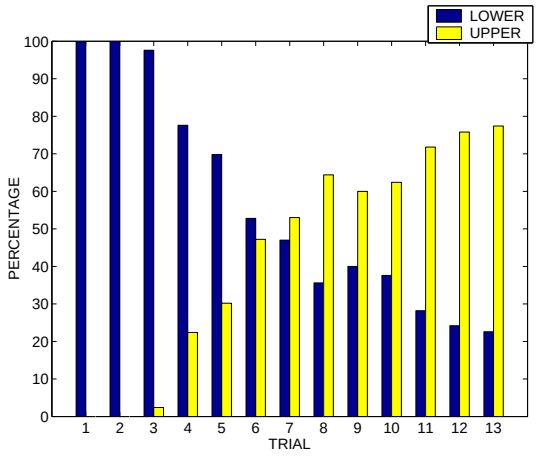
compared to the single equation results, shown in Figure 4.3, while the percentages and mean final values for trials 1 through 5 are similar to those for the single differential equation, Figure 4.1.

Comparing the results for the second set of initial conditions for the full system, Table 4.5, and the single equation, Figure 4.3, for the second set of initial conditions shows close agreement for trials 1 through 7 of the percentage of X final path values above the middle critical value. The percentages are not exactly the same but are close enough to determine that the switching behavior is similar, though the mean final values, Figure 4.1 for the single equation, are not necessarily alike. The mean values for the full system are significantly smaller for trials 3, 4, and 5 while the value for trial 7 is larger than for the single equation.

For the third set of initial conditions, trials 4 through 7 for the single equation, Figure 4.4, and the full equations, Table 4.6, show similar percentages of final path values near the upper critical value. For trials 1 through 3 the single percentages show a significant amount of switching to the lower critical value, however, the X final path values for the full system are almost all near the upper critical value.



(a) % of final time path values near lower or upper stable critical value, initial condition 1, 500 paths used, σ^2 values for the trials are presented in Table 4.4



(b) % of final time path values near lower or upper stable critical value, initial condition 2, 500 paths used, σ^2 values for the trials are presented in Table 4.4

Figure 4.3: Euler-Maruyama Percentages for Single SDE, Additive Noise, Initial Conditions 1 and 2

For the fourth set of initial conditions the single equation, Figure 4.4, showed no switching behavior for trials 1 through 7, while the full system, Table 4.6, had a significant amount of switching for trials 4 through 7. As a result the final mean values for the full system are smaller than for the single equation, Figure 4.1, for trials 4 through 7.

As σ^2 tended to zero the final average values tended to the deterministic results, obtained using the backward Euler method, for all the initial conditions. It took much smaller σ^2 values to cause the switching for the full system than it did for the single equation. It's easy to see why the larger σ^2 values had a more dramatic effect on the full system than they did on the single differential equation. In the full system each species was being perturbed by noise and the perturbation of one species affects the value of each of the others, thus causing a cumulative effect of the noise on any one specie's value.

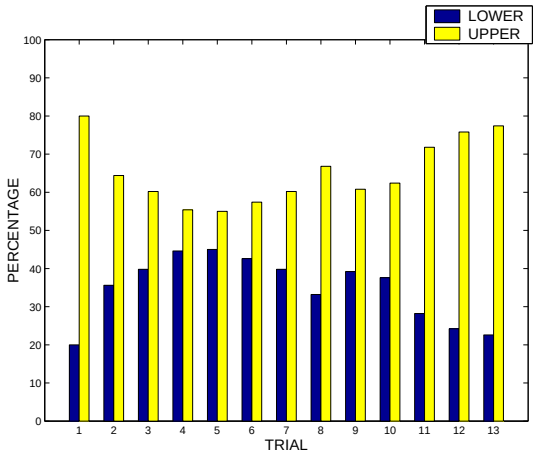
4.2.1.2 Multiplicative Noise. The mean final X values for the single differential equation (2.10) with multiplicative noise, of the form $g(x) = x\sqrt{\sigma^2}$, are

Table 4.5: Euler-Maruyama Mean Final Values for Full System, Additive Noise, Initial Conditions 1 and 2, 500 Paths Used

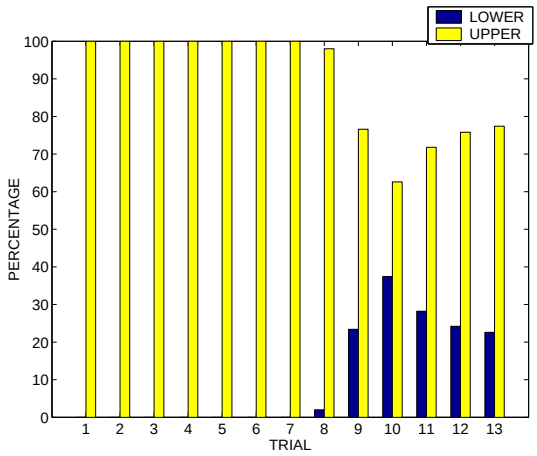
	Initial Conditions 1					Initial Conditions 2				
trial	% upper	X	X_2	D	DX_2	% upper	X	X_2	D	DX_2
7	48.8	0.3548	1.7042	1.6352	0.3308	56.4	0.6608	3.1764	1.3336	0.6324
6	33.6	0.1242	0.3657	1.8892	0.0868	47.4	0.5202	2.1082	1.4935	0.4825
5	9.6	0.0460	0.0054	1.9835	0.0057	22.2	0.2802	0.8503	1.7490	0.2402
4	3.4	0.0451	0.0047	1.9875	0.0049	12.8	0.1874	0.4738	1.8451	0.1473
3	0.0	0.0442	0.0041	1.9924	0.0042	0.4	0.0501	0.0223	1.9865	0.0101
2	0.0	0.0441	0.0040	1.9936	0.0040	0.0	0.0441	0.0040	1.9936	0.0040
1	0.0	0.0439	0.0039	1.9950	0.0039	0.0	0.0439	0.0039	1.9950	0.0039

Table 4.6: Euler-Maruyama Mean Final Values for Full System, Additive Noise, Initial Conditions 3 and 4, 500 Paths Used

	Initial Conditions 3					Initial Conditions 4				
trial	% upper	X	X_2	D	DX_2	% upper	X	X_2	D	DX_2
8	-	-	-	-	-	71.2	1.5207	8.6217	0.4951	1.4629
7	59.6	0.8475	4.0748	1.1481	0.8179	61.2	0.9036	4.3305	1.0926	0.8734
6	54.4	0.7624	3.1073	1.2517	0.7242	57.0	0.8631	3.5795	1.1496	0.8264
5	57.6	0.8361	2.6322	1.1940	0.7953	63.0	0.9177	2.8784	1.1123	0.8769
4	66.2	0.9293	2.6959	1.1045	0.8879	71.2	0.9903	2.8669	1.0443	0.9481
3	91.4	1.1622	3.0249	0.8764	1.1202	93.6	1.1857	3.0772	0.8536	1.1430
2	98.0	1.2349	3.1692	0.8038	1.1938	98.4	1.2435	3.1903	0.7953	1.2023
1	100.0	1.2799	3.2853	0.7589	1.2401	100.0	1.2804	3.2877	0.7585	1.2404



(a) % of final time path values near lower or upper stable critical value, initial condition 3, 500 paths used, σ^2 values for the trials are presented in Table 4.4



(b) % of final time path values near lower or upper stable critical value, initial condition 4, 500 paths used, σ^2 values for the trials are presented in Table 4.4

Figure 4.4: Euler-Maruyama Percentages for Single SDE, Additive Noise, Initial Conditions 3 and 4

shown graphically in Figures 4.5 and 4.6. For ease of comparison, the mean values obtained using additive noise are also presented. The percentages of final time path values that were near the upper or lower critical values are graphed for the different trials in Figures 4.7 and 4.8. The procedure followed is described in section 3.2.1.

Almost no switching to the upper critical value occurred for the first set of initial conditions with multiplicative noise, shown in Figure 4.7. This is very different from the case of additive noise, in which a significant amount of switching took place for trials 8 through 13, Figure 4.3. Consequently, as seen in Figure 4.5, the mean values for multiplicative noise are all very close to the lower critical value while the mean values for additive noise with the larger σ^2 values are significantly larger. Species X in the first set of initial conditions, given in Table 4.3, has an initial value less than 1 and as a result the noise term for multiplicative noise is smaller than that for additive noise with the same σ^2 value.

For the second set of initial conditions, shown in Figure 4.7, the only significant switching across the middle critical value took place for trials 4 through 9. No trial

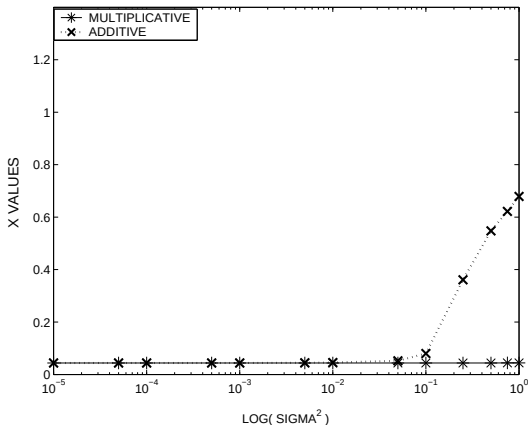
had a majority of final path values near upper the critical value, while for additive noise trials 7 through 13 did, Figure 4.3. The mean values for multiplicative noise were significantly smaller than for additive noise for most σ^2 values, shown in Figure 4.7.

For additive noise, Figure 4.4, the majority of the final values for the third set of initial conditions ended near the upper critical value for all the trials, but with the multiplicative noise the upper critical value only dominates for trials 1 through 5 as shown in Figure 4.8. Almost all the values switched across the middle critical value for trials 10 through 13. For the larger σ^2 values the mean values for multiplicative noise are significantly smaller than for additive noise, as seen in Figure 4.6, but for the smaller σ^2 values with multiplicative noise the mean values are larger than for the additive noise.

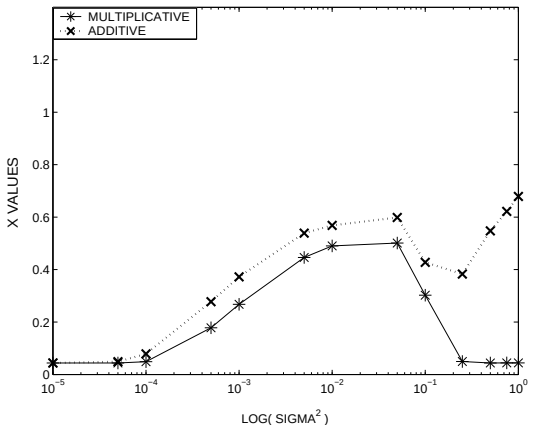
Comparing the additive and multiplicative results for the fourth set of initial conditions, given in Figure 4.6, shows similar mean values for the smaller σ^2 values. The majority of final path values for multiplicative noise, Figure 4.8, end up almost exclusively near the lower critical value for trials 10 through 13, while the additive noise results, shown in Figure 4.4, indicate most final path values are near the upper critical value.

As σ^2 tended to zero the final average values tended to the deterministic results, obtained using the forward Euler method, for all the initial conditions. While the final path values for the additive noise showed a propensity to end up near the upper critical value for the larger σ^2 values of trials 10 through 13, the results for multiplicative noise show a tendency toward the lower critical value for these same trials.

No problems of the inverted matrix becoming nearly singular were encountered when solving the full system with multiplicative noise. The form of the multiplicative noise term added to each species is $\sqrt{\sigma^2}$ multiplied by the value of that particular species. Each species except D in the first two sets of initial conditions, given in



(a) mean final X values initial condition 1: x = 0.37, for 25 time units, 500 paths used, critical values: 0.0438, 0.7088, 1.2873

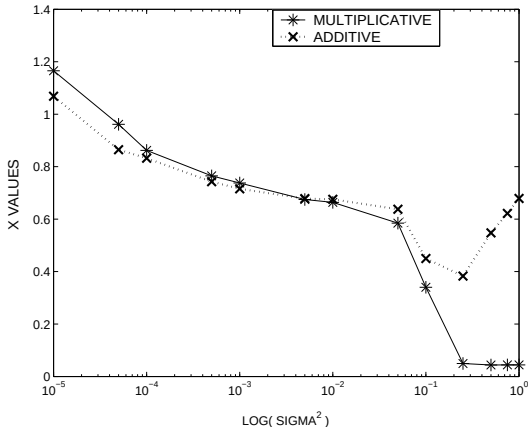


(b) mean final X values initial condition 2: x = 0.7, for 25 time units, 500 paths used, critical values: 0.0438, 0.7088, 1.2873

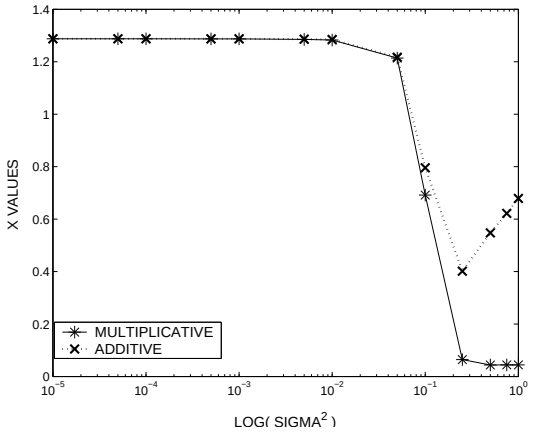
Figure 4.5: Euler-Maruyama Mean Values for Single SDE, Multiplicative Noise, Initial Conditions 1 and 2

Table 4.3, has an initial value less than 1 and as a result the noise term added to all species other than D for multiplicative noise is smaller than that for additive noise with the same σ^2 value. The X value for the third and fourth initial condition sets are also less than one. This reduction in the magnitude of some of the noise terms may be the reason that no errors indicating singularity of the inverted matrix were encountered. The mean final values of all species for the first and second set of initial conditions are given in Table 4.7 and in Table 4.8 for the third and fourth set of initial conditions. The column under the "% upper" heading indicates the percentage of X final time path values that were greater than or equal to the middle critical value. The procedure followed is the same as for the single differential equation presented in section 3.2.1, except that the calculations are carried out for 50 time units and averages are calculated for each species.

The final mean values for the first set of initial conditions, Table 4.7, for trials 8 through 12 are larger than for the single equation results, Figure 4.5, but the percentages for the final path values of all trials are very close as a small amount



(a) mean final X values initial condition 3: x = 0.71, for 25 time units, 500 paths used, critical values: 0.0438, 0.7088, 1.2873



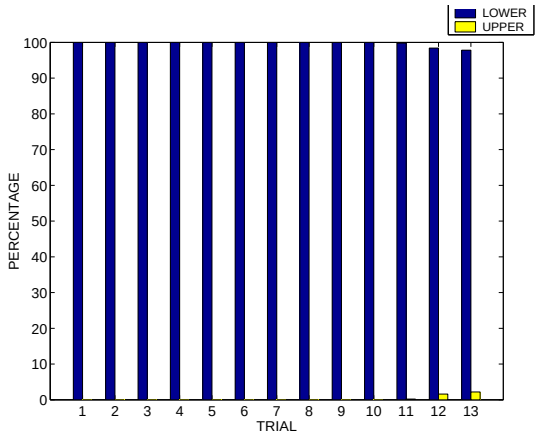
(b) mean final X values initial condition 4: x = 0.95, for 25 time units, 500 paths used, critical values: 0.0438, 0.7088, 1.2873

Figure 4.6: Euler-Maruyama Mean Values for Single SDE, Multiplicative Noise, Initial Conditions 3 and 4

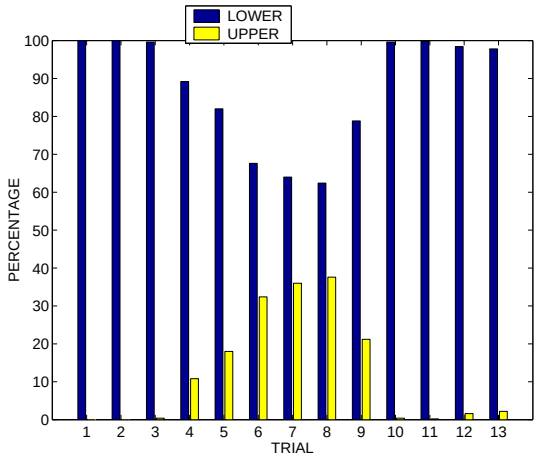
of switching behavior occurs with the full system and almost none with the single equation, Figure 4.7.

For the second set of initial conditions the single equation, Figure 4.7, shows slightly higher percentages for some of the trials, when compared to the full system, Table 4.7. However, the mean values for the full system are larger than for the single equation, shown in Figure 4.5, for most of the trials. This is observed for trials 7, 8, and 9.

Trials 10 through 13 for the single equation, Figure 4.8, and the full system, Table 4.8, with the third set of initial conditions show similar switching behavior. Hardly any final path values are near the upper critical value. The single equation shows more switching to the lower critical value for trials 1 through 5 than the full system does. As with the second set of initial conditions, trials 7, 8, and 9 have smaller percentages for final path values near the upper critical value for the full system, yet the full system has larger mean values than for the single equation, Figure 4.6.



(a) % of final time path values near lower or upper stable critical value, initial condition 1, 500 paths used, σ^2 values for the trials are presented in Table 4.4



(b) % of final time path values near lower or upper stable critical value, initial condition 2, 500 paths used, σ^2 values for the trials are presented in Table 4.4

Figure 4.7: Euler-Maruyama Percentages for Single SDE, Multiplicative Noise, Initial Conditions 1 and 2

For the fourth set of initial conditions the full system, Table 4.8, shows close agreement with the single equation, Figure 4.8, for trials 10 through 13. The full system exhibits much more switching behavior to the lower critical value than the single equation does for trials 4 through 7 and as a result the X mean values for the full system are smaller than for the single equation, 4.6.

As σ^2 tended to zero the final average values tended to the deterministic results, obtained using the backward Euler method, for all the initial conditions. Just like the single equation, the final path values for the full system tended to the lower critical value for the larger σ^2 values of trials 10 through 13.

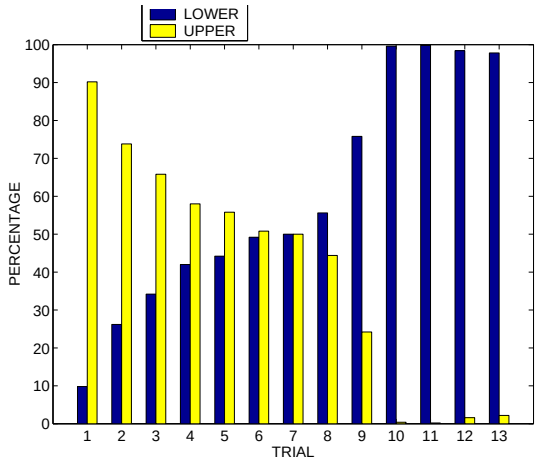
4.2.1.3 Weak Convergence. The parameter values in Table 4.1 with the X initial condition from set 4 in Table 4.3 were used for the calculations. For both procedures the σ^2 values given in Table 4.4 were used to vary the noise strength. Calculations were started at $\sigma^2 = 1$ and then decreased until the plots of the errors were somewhat parallel to the reference line of slope one. The slope of the error

Table 4.7: Euler-Maruyama Mean Final Values for Full System, Multiplicative Noise, Initial Conditions 1 and 2, 500 Paths Used

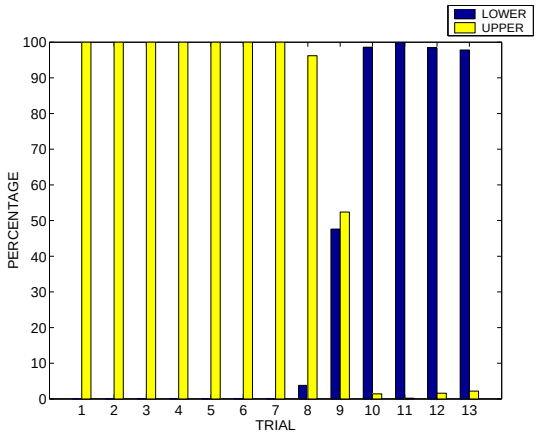
	Initial Conditions 1					Initial Conditions 2				
trial	% upper	X	X_2	D	DX_2	% upper	X	X_2	D	DX_2
13	0.0	0.0399	0.0033	0.0003	0.0000	0.0	0.0399	0.0033	0.0005	0.0000
12	0.2	0.0562	0.2543	0.0037	0.0259	0.4	0.0572	0.2641	0.0056	0.0284
11	1.4	0.2178	10.2855	0.0382	0.3174	0.8	0.1436	5.1617	0.0291	0.2989
10	5.0	0.6379	40.9545	0.2418	0.9528	4.6	0.6818	39.9750	0.2143	1.0997
9	10.0	1.0206	42.6589	0.5641	1.4596	10.8	0.9324	37.9724	0.6003	1.2590
8	10.8	0.7688	17.4066	1.0149	0.9866	13.8	0.8055	16.9373	0.9677	0.9549
7	3.8	0.1602	1.8176	1.8475	0.1503	18.2	0.5321	3.9968	1.4984	0.5227
6	0.6	0.0500	0.0282	1.9610	0.0114	19.6	0.4437	2.0970	1.6031	0.4076
5	0.0	0.0439	0.0039	1.9881	0.0039	11.6	0.2349	0.7198	1.8006	0.1948
4	0.0	0.0438	0.0038	1.9909	0.0038	8.0	0.1593	0.3911	1.8760	0.1189
3	0.0	0.0438	0.0038	1.9940	0.0038	0.0	0.0438	0.0038	1.9931	0.0038
2	0.0	0.0438	0.0038	1.9947	0.0038	0.0	0.0438	0.0038	1.9941	0.0038
1	0.0	0.0438	0.0038	1.9955	0.0038	0.0	0.0438	0.0038	1.9952	0.0038

Table 4.8: Euler-Maruyama Mean Final Values for Full System, Multiplicative Noise, Initial Conditions 3 and 4, 500 Paths Used

	Initial Conditions 3					Initial Conditions 4				
trial	% upper	X	X_2	D	DX_2	% upper	X	X_2	D	DX_2
13	0.0	0.0399	0.0033	0.0007	0.0000	0.0	0.0399	0.0033	0.0008	0.0000
12	0.2	0.0555	0.2984	0.0057	0.0298	0.2	0.0561	0.3217	0.0071	0.0317
11	0.8	0.1760	8.0533	0.0274	0.3146	0.8	0.2053	12.9450	0.0280	0.3220
10	4.2	0.7080	67.9007	0.2209	0.9793	3.6	0.5955	31.8561	0.2213	0.8570
9	11.2	0.8705	30.3146	0.6104	1.1997	11.2	0.8918	31.4998	0.5859	1.2062
8	15.6	0.8277	16.6442	0.9267	0.9693	15.6	0.8414	17.3453	0.8994	0.9740
7	26.8	0.6899	4.7832	1.3276	0.6708	29.2	0.7135	4.5062	1.2611	0.6874
6	33.2	0.6990	3.2863	1.3010	0.6651	36.8	0.7623	3.5775	1.2454	0.7290
5	52.8	0.8140	2.6015	1.2174	0.7742	57.8	0.8870	2.8371	1.1433	0.8471
4	65.4	0.9208	2.6862	1.1148	0.8795	70.6	0.9779	2.8422	1.0592	0.9360
3	91.8	1.1660	3.0347	0.8743	1.1243	94.0	1.1834	3.0707	0.8571	1.1409
2	97.8	1.2370	3.1768	0.8024	1.1962	98.6	1.2423	3.1873	0.7968	1.2013
1	100.0	1.2803	3.2873	0.7587	1.2404	100.0	1.2802	3.2870	0.7587	1.2403



(a) % of final time path values near lower or upper stable critical value, initial condition 3, 500 paths used, σ^2 values for the trials are presented in Table 4.4



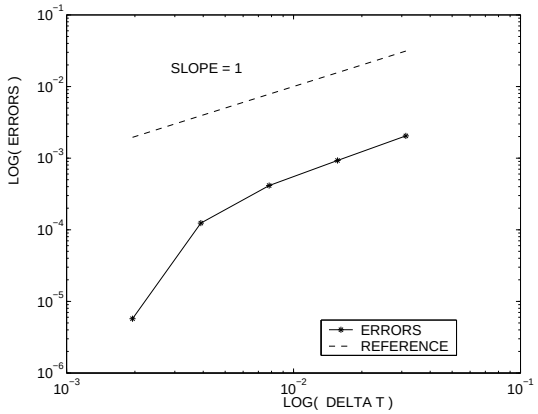
(b) % of final time path values near lower or upper stable critical value, initial condition 4, 500 paths used, σ^2 values for the trials are presented in Table 4.4

Figure 4.8: Euler-Maruyama Percentages for Single SDE, Multiplicative Noise, Initial Conditions 3 and 4

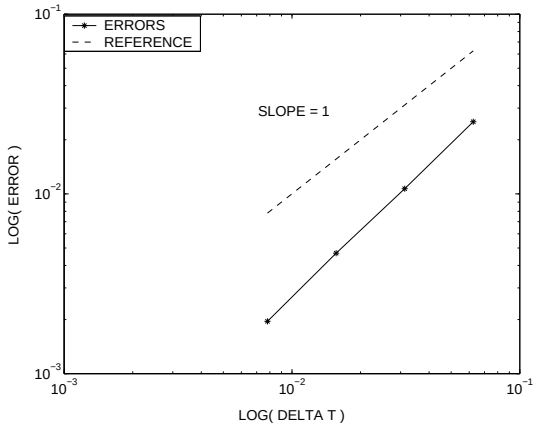
plot is referred to as γ. It is the approximated order of convergence, as described in section 3.2.1.1.

For the first procedure using additive noise, the errors plotted as almost horizontal lines with all calculated γ values being negative and very close to zero for σ^2 values of 1 through 0.005. Using σ^2 values of 0.001 and 0.0005 showed improvement with γ values of around 0.26 and 0.74. Using $\sigma^2 = 0.0001$ gave even better γ values and an error plot is shown in Figure 4.9. Even though $\gamma = 1.9867$, it's clear that this is not a straight line with that slope. A better than expected error has improved on the expected γ value of 1. This results indicate that the expected value of the SDE (3.1) with additive noise, as described in section 3.2.1.1, is not the deterministic solution.

The second procedure of 3.2.1.1 produced a γ value of 1.2262 with $\sigma^2 = 1$. It was not necessary to continue with decreasing σ^2 values. The error plot is shown in Figure 4.9.



(a) procedure 1 used, errors plotted using the deterministic value as the true expected value, calculated order of convergence of $\gamma = 1.9867$ using $\sigma^2 = 0.0001$



(b) procedure 2 used, errors plotted using a path of $\Delta t = 2^{-11}$ to approximate the true expected value, calculated order of convergence of $\gamma = 1.2262$ using $\sigma^2 = 1$,

Figure 4.9: Weak convergence of the Euler-Maruyama method for the single SDE with additive noise g(x) = $\sqrt{\sigma^2}$,

For the second procedure using multiplicative noise, σ^2 values of 1 and 0.5 produced widely varying γ values depending on the seed value for the random number generator. The γ's ranged from negative values to being close to 1. Using $\sigma^2 = 0.5$ gave γ values between 0 and 1.25 while for $\sigma^2 = 0.25$, the γ values were between 0.65 and 1.35. Running the procedure ten time with different seed values with $\sigma^2 = 0.25$ resulted in three with γ values less than 0.9. The cutoff value of 0.9 was used since it is ten percent relative error away from the desired γ value of 1. With $\sigma^2 = 0.05$, only one out of the ten different times the procedure was carried out with a different seed values produced a γ less than 0.9. Since the convergence calculations seemed only slightly dependent on the Wiener paths sampled with $\sigma^2 = 0.05$, the process was stopped. An error plot using $\sigma^2 = 0.05$ is shown in Figure 4.10.

4.2.1.4 Strong Convergence. The parameter values in Table 4.1 with the X initial condition from set 4 in Table 4.3 were used for the calculations. The procedure is described in section 3.2.1.2. The σ^2 values given in Table 4.4 were used to vary the noise strength. Calculations were started at $\sigma^2 = 1$ and then decreased

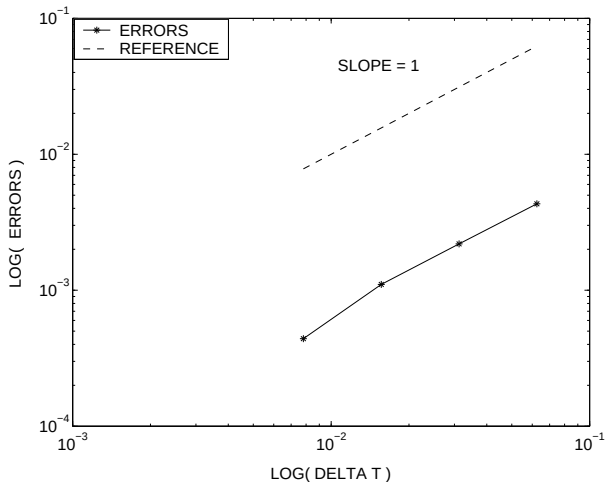


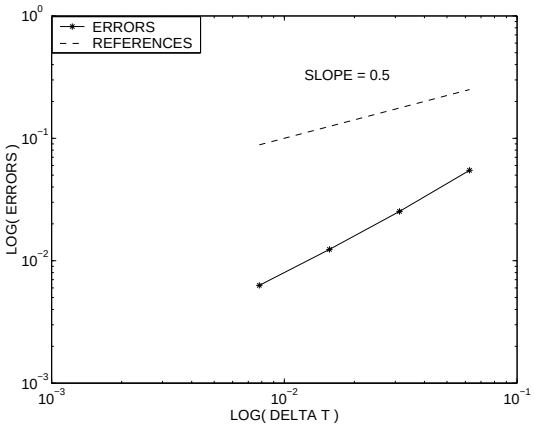
Figure 4.10: Weak convergence of the Euler-Maruyama method for the single SDE with multiplicative noise g(x) = x$\sqrt{\sigma^2}$, procedure 2 used, errors plotted using a path of $\Delta t = 2^{-11}$ to approximate the true expected value, calculated order of convergence of $\gamma = 1.0870$ using $\sigma^2 = 0.05$

until the plots of the errors were somewhat parallel to the reference line of slope one half. Just as with the weak convergence, the slope of the error plot is referred to as γ. It is the approximated order of convergence, as described in section 3.2.1.1.

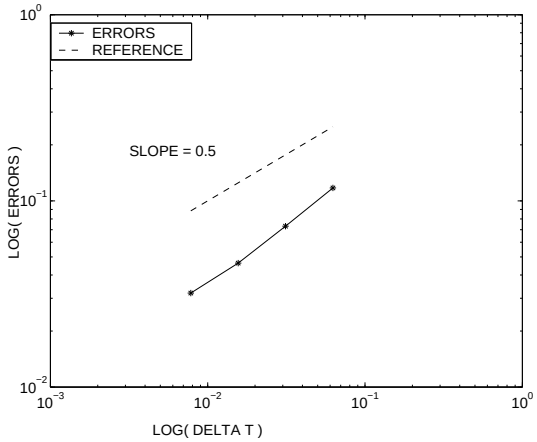
With additive noise, using $\sigma^2 = 1$ resulted in a γ value of 1.0403. This is much better than the theoretical order of convergence of 0.5. Changing the random number generator consistently produced γ values close to 1. The error plot is shown in Figure 4.11.

Using multiplicative noise produced a γ value of 0.6563 with $\sigma^2 = 1$. It was not necessary to continue with decreasing σ^2 values since the results were consistent when using different seed values for the random number generator. This is a significant amount better than the theoretical $\gamma = 0.5$. The error plot is shown in Figure 4.11.

4.2.2 Milstein's Method. The explicit Milstein method was used to solve the single SDE. The results using Milstein's method were very close to those obtained using the Euler-Maruyama method with multiplicative noise. Taking the difference of the results for both methods gave very small values. In fact the plots for the



(a) additive noise, errors plotted using a path of $\Delta t = 2^{-11}$ to approximate the true expected value, calculated order of convergence of $\gamma = 1.0403$ using $\sigma^2 = 1$



(b) multiplicative noise, errors plotted using a path of $\Delta t = 2^{-11}$ to approximate the true expected value, calculated order of convergence of $\gamma = 0.6563$ using $\sigma^2 = 1$

Figure 4.11: Strong convergence of the Euler-Maruyama method for the single SDE with additive noise g(x) = $\sqrt{\sigma^2}$ and multiplicative noise g(x) = x$\sqrt{\sigma^2}$

Milstein results are not presented because they were the exact same as Figures 4.5 and 4.6 for the EM results.

The implicit Milstein method was employed for the full system. The procedure is described in section 3.2.2. The $f(\vec{x}_{i+1})$ term in equation (3.10) was expanded in a Taylor series as shown in equation (4.4). The following form was used for calculations:

$$\vec{x}_{i+1} = (1 - \triangle t J(\vec{x}_i))^{-1}(\vec{x}_i + \triangle t f(\vec{x}_i) - \triangle t J(\vec{x}_i)\vec{x}_i + g(\vec{x}_i)\overrightarrow{\triangle W}_{i+1} + \frac{1}{2}g(\vec{x}_i)\vec{\alpha}) \tag{4.9}$$

where the kth component of $\vec{\alpha}$ is defined as $\sqrt{\sigma^2}((\overrightarrow{\triangle W}^k_{i+1})^2 - \Delta t)$.

The results for the full system are presented in Tables 4.9 and 4.10. When these final values are compared with the final values obtained using the EM method for the full system, in Tables 4.7 and 4.8, they are seen to be quite close for all species and all σ^2 values.

Table 4.9: Milstein Mean Final Values for Full System, Multiplicative Noise, Initial Conditions 1 and 2, 500 Paths Used

	Initial Conditions 1					Initial Conditions 2				
trial	% upper	X	X_2	D	DX_2	% upper	X	X_2	D	DX_2
13	0.0	0.0399	0.0033	0.0003	0.0000	0.0	0.0403	0.0040	0.0009	0.0001
12	0.2	0.0570	0.2542	0.0038	0.0252	0.4	0.0574	0.2614	0.0056	0.0288
11	1.4	0.2082	9.1635	0.0379	0.2989	0.8	0.1389	4.7369	0.0291	0.2833
10	5.0	0.6289	39.4227	0.2419	0.9431	4.2	0.6731	39.2981	0.2117	1.0757
9	10.0	1.0068	41.8180	0.5698	1.4515	11.0	0.9297	37.6273	0.6009	1.2548
8	10.8	0.7682	17.3858	1.0137	0.9877	14.0	0.8039	16.8640	0.9669	0.9525
7	3.8	0.1598	1.8055	1.8472	0.1498	18.2	0.5323	3.9960	1.4983	0.5229
6	0.6	0.0501	0.0286	1.9607	0.0115	19.6	0.4437	2.0963	1.6029	0.4076
5	0.0	0.0439	0.0039	1.9881	0.0039	11.6	0.2349	0.7198	1.8006	0.1948
4	0.0	0.0438	0.0038	1.9908	0.0038	8.0	0.1591	0.3903	1.8762	0.1187
3	0.0	0.0438	0.0038	1.9940	0.0038	0.0	0.0438	0.0038	1.9931	0.0038
2	0.0	0.0438	0.0038	1.9947	0.0038	0.0	0.0438	0.0038	1.9941	0.0038
1	0.0	0.0438	0.0038	1.9955	0.0038	0.0	0.0438	0.0038	1.9952	0.0038

Table 4.10: Milstein Mean Final Values for Full System, Multiplicative Noise, Initial Conditions 3 and 4, 500 Paths Used

	Initial Conditions 3					Initial Conditions 4				
trial	% upper	X	X_2	D	DX_2	% upper	X	X_2	D	DX_2
13	0.0	0.0399	0.0033	0.0009	0.0000	0.0	0.0399	0.0033	0.0010	0.0000
12	0.2	0.0544	0.2587	0.0060	0.0269	0.2	0.0546	0.2663	0.0072	0.0275
11	1.0	0.1694	6.9940	0.0299	0.3020	0.8	0.1962	12.3750	0.0279	0.3070
10	4.0	0.6967	62.9397	0.2205	0.9746	3.6	0.7035	62.4842	0.2228	0.9425
9	11.2	0.8728	30.2816	0.6071	1.2017	10.8	0.8918	31.3972	0.5841	1.2045
8	15.6	0.8240	16.5771	0.9292	0.9641	15.6	0.8401	17.2907	0.8999	0.9731
7	26.8	0.6902	4.7850	1.3274	0.6711	29.4	0.7138	4.5064	1.2609	0.6877
6	33.4	0.6996	3.2924	1.3009	0.6657	36.4	0.7541	3.5120	1.2509	0.7198
5	53.0	0.8166	2.6084	1.2148	0.7768	57.8	0.8869	2.8370	1.1433	0.8471
4	65.4	0.9208	2.6862	1.1148	0.8795	70.6	0.9779	2.8421	1.0593	0.9360
3	91.8	1.1660	3.0347	0.8744	1.1243	94.0	1.1834	3.0707	0.8571	1.1409
2	97.8	1.2370	3.1768	0.8024	1.1962	98.6	1.2423	3.1872	0.7968	1.2013
1	100.0	1.2803	3.2873	0.7587	1.2404	100.0	1.2802	3.2870	0.7587	1.2403

4.2.2.1 Strong Convergence. The parameter values in Table 4.1 with the X initial condition from set 4 in Table 4.3 was used for the calculations. The procedure is described in section 3.2.2.1. The σ^2 values given in Table 4.4 were used to vary the noise strength. Calculations were started at $\sigma^2 = 1$ and then decreased until the plots of the errors were somewhat parallel to the reference line of slope one half.

Using $\sigma^2 = 1$ resulted in a $\gamma = 1.2070$. The γ values remained consistent when using different seed values for the random number generator. Thus, it was not necessary to continue with decreasing σ^2 values. The error plot is shown in Figure 4.12.

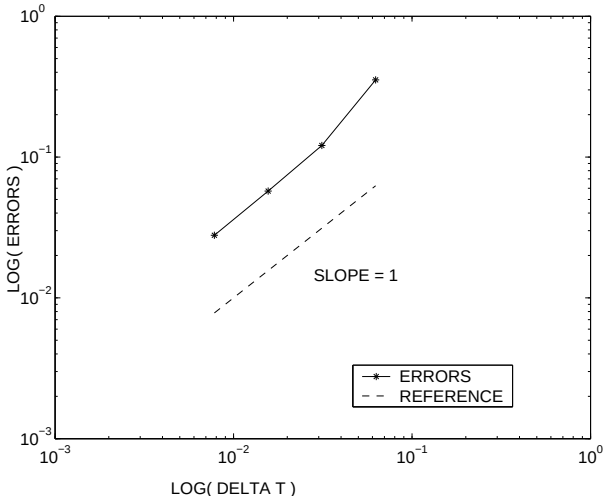
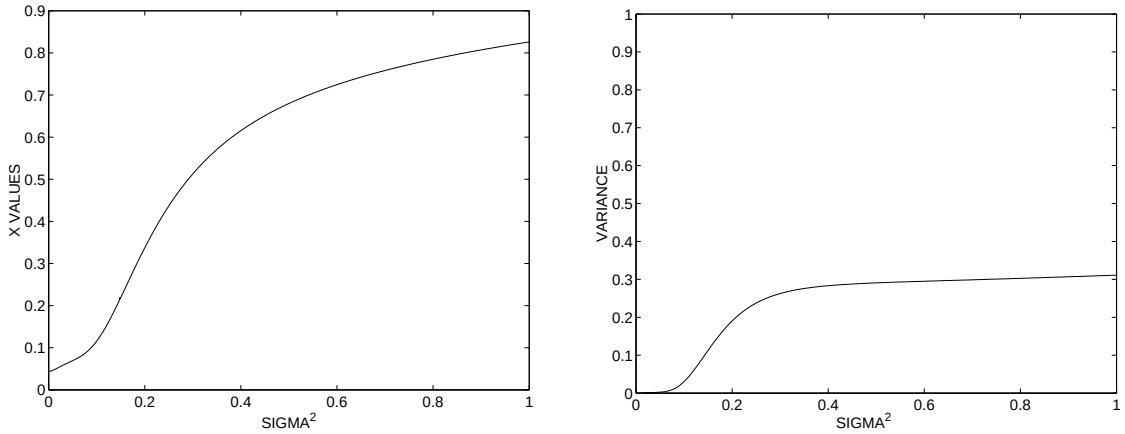


Figure 4.12: Strong convergence of the Milstein method for the single SDE with multiplicative noise g(x) = x$\sqrt{\sigma^2}$, errors plotted using a path of $\Delta t = 2^{-11}$ to approximate the true value, calculated order of convergence of $\gamma = 1.2070$ using $\sigma^2 = 1$

4.2.3 Fokker-Planck. The Fokker-Planck equation was solved in Mathcad for values of σ^2 from 0.001, 0.002, ... 1. The function $\phi(X)$, equation (3.22), was integrated using Mathcad's adaptive quadrature function. The mean integral, equation (3.23), and the normalizing factor A, equation (3.20), were calculated using Mathcad's infinite integral function. The variance was also calculated for each σ^2

value. The results are shown in figure 4.13. The procedure is described in section 3.2.3.

Increasing the σ^2 value caused the mean values to increase. This is similar to the behavior seen with the Euler-Maruyama method for additive noise, shown in Figures 4.1 and 4.2. However, for smaller σ^2 values, the EM results depended on the initial condition used.



(a) Fokker-Planck mean steady state X value (b) Fokker-Planck steady state variances

Figure 4.13: Fokker-Planck Steady State Mean X Values and Variances

4.2.4 GRASS. The results from the procedure at the end of section 3.2.4 are presented in two tables. Table 4.11 contains the mean final values for the proteins and their dimers. Table 4.12 contains the mean final values for the DNA sites and the different binding permutations that involve the proteins.

Three deterministic results of the system are reported using GRASS, the ode23s function in Matlab, and the backward Euler method. The differences between the GRASS and ode23s results are negligible. Using the backward Euler method resulted in larger errors, but even then the largest relative error when compared to the values obtained with the ode23s function is 1.5% for the $P_{2,\,2}$ species.

The stochastic mean final values for GRASS are close to the deterministic solution for all species. The backward Euler form given by equation (4.8) was used

Table 4.11: GRASS Protein Mean Final Values, 20 Paths Used

		$P_{1,1}$	$P_{2,1}$	$P_{1,2}$	$P_{2,2}$
deterministic	GRASS	45.5379	31.0979	96.9017	69.0773
	ode23s	45.5379	31.0980	96.9017	69.0774
	backward Euler	45.4408	30.8629	96.4892	68.0369
stochastic	GRASS	45.0720	30.8435	97.0928	69.7751
	Euler-Maruyama	41.8070	29.2312	85.5210	75.7929
	Stochastica	62.3900	43.8796	90.0720	68.7988

Table 4.12: GRASS DNA Mean Final Values, 20 Paths Used

		$D_{1,0}$	$D_{1,1}$	$D_{1,2}$	$D_{2,0}$	$D_{2,1}$	$D_{2,2}$
deterministic	GRASS	0.5953	28.8434	20.5613	0.5953	28.8434	20.5613
	ode23s	0.5953	28.8434	20.5613	0.5953	28.8434	20.5613
	backward Euler	0.6005	28.9712	20.4283	0.5959	28.7484	20.2712
stochastic	GRASS	0.5647	28.5930	20.8423	0.6037	28.8909	20.5054
	Euler-Maruyama	0.8001	36.5643	15.9401	0.5950	24.4265	21.8593
	Stochastica	0.6270	28.3582	21.0148	0.6200	27.6902	21.6898

for the EM calculations with the diagonal additive noise form as described at the end of section 3.2.1. The EM mean final values for $P_{2,1}$, $D_{2,0}$, and $D_{2,2}$ are significantly further away from the deterministic values. The Stochastica mean final values for the $P_{1,1}$ and $P_{2,1}$ species are significantly larger than the deterministic values. The description of Stochastica is given in section 3.3.1. Reproduction of the GRASS results are hampered by the fact that no information is given as to how the noise is added to the system by the SDE solver. The mean paths for the different approaches are shown in Figure 4.14. A single path from each approach is shown in Figure 4.15.

In the spirit of Hasty et al. and Campbell, the system of ten differential equations presented in section 3.2.4 can be easily reduced to four differential equations by assuming that reactions (3.36) through (3.44) are much slower than reactions (3.45) through (3.50). Because none of the DNA species values are altered in reactions (3.45) through (3.50), the differential equations for the DNA species can be assumed to reach steady state much quicker than the differential equations for the different protein species. Thus the state of the system can be studied with a 60% reduction in

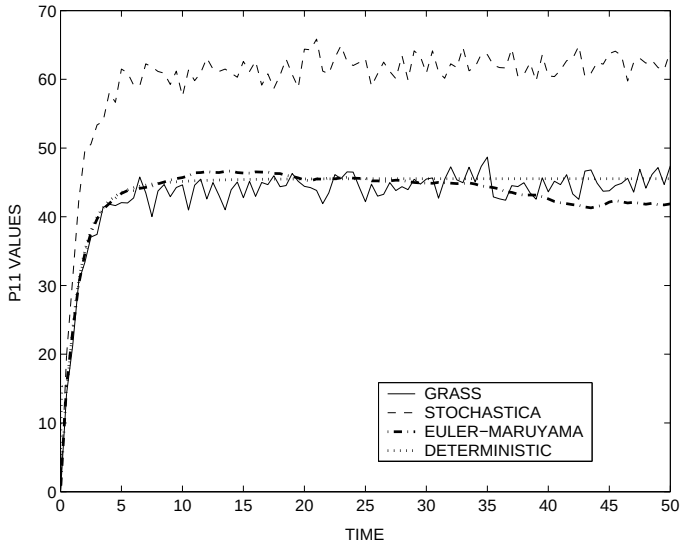


Figure 4.14: GRASS: P11 values for the mean path, 20 paths used, paths are plotted at 0.5 time unit intervals

the complexity of the calculations. With four algebraic equations, finding the critical values is more difficult than for the Hasty et al. protein regulation model and this can be a topic for future work.

4.3 Gillespie's Exact Method

Probably the best properties of Gillespie's exact method are that it is easy to understand and easy to implement. The method does not require any knowledge or experience with solving systems of differential equations, though a basic understanding of probability distributions would probably be necessary. The method was very easy to write a Matlab script for. This was done to reproduce the exact method calculations that Stochastica and ESS perform. A problem with the method is that it only uses nonnegative integer values for the different species. This might cause a problem for examining systems in which the different critical values are less than one. With the differential equation approach the system could be perturbed by any desired small value. The Gillespie approach does not have this capability. The different sets of parameter values used for the Hasty protein regulation model solved

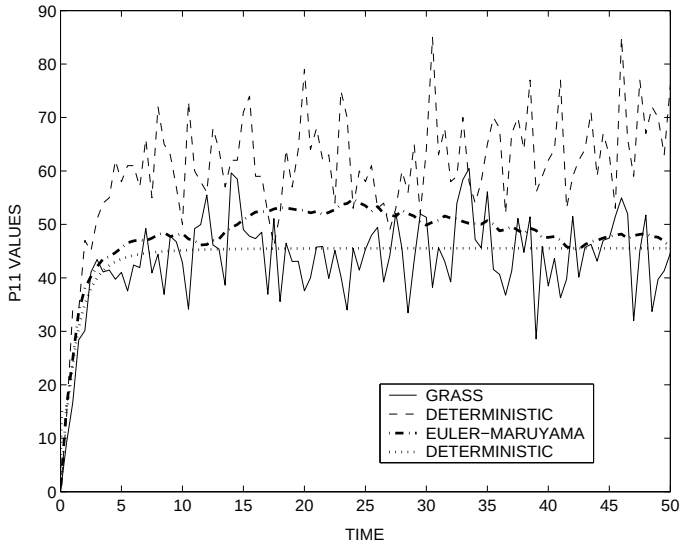


Figure 4.15: GRASS: P11 values for a single path, paths are plotted at 0.5 time unit intervals

with Stochastica, ESS, the Matlab implementation of the direct method, and the τ-leaping method are given in Table 4.13. The first set of parameter values was chosen because there is only one real critical value. The critical values for each parameter set are given in Table 4.14. The initial conditions are given in Table 4.15.

4.3.1 Stochastica. The mean final values for the Stochastica calculations, obtained following the procedure in section 3.3.1, are given in Table 4.16. The Matlab Direct entries refer to the direct method implementation of the exact method, presented in section 2.3.3, that was done in Matlab.

Table 4.13: Model Parameter Values Used for Stochastica, ESS, and τ-Leaping

	Trials 1-2	Trials 3-4	Trials 5-7
n	3	3	2
p	2	2	2
r	0.4	0.4	0.4
k_1	1	10	10
k_{-1}	1	5	5
k_2	1	5	5
k_{-2}	1	10	10
k_3	4	5	2.5
k_4	10	10	10
d_T	23	2	2

Table 4.14: Resulting Critical Values of the Model Used for Stochastica, ESS, and τ-Leaping

Trials		X	X_2	D	DX_2
1-2	Stable	55.2219	3049.5	0.0075	22.9925
		0.0090 + 0.0253i	-5.606E-4 + 4.587E-4i	23.0129 - 0.0106i	-0.0129 + 0.0106i
		0.0090 - 0.0253i	-5.606E-4 - 4.587E-4i	23.0129 + 0.0106i	-0.0129 - 0.0106i
3-4	Stable	5.8708	68.9332	0.0564	1.9436
	Unstable	0.1031	0.0212	1.9790	0.0210
	Stable	0.0661	0.0087	1.9913	0.0087
5-7	Stable	1.2873	3.3145	0.7527	1.2473
	Unstable	0.7088	1.0049	1.3312	0.6688
	Stable	0.0438	0.0038	1.9962	0.0038

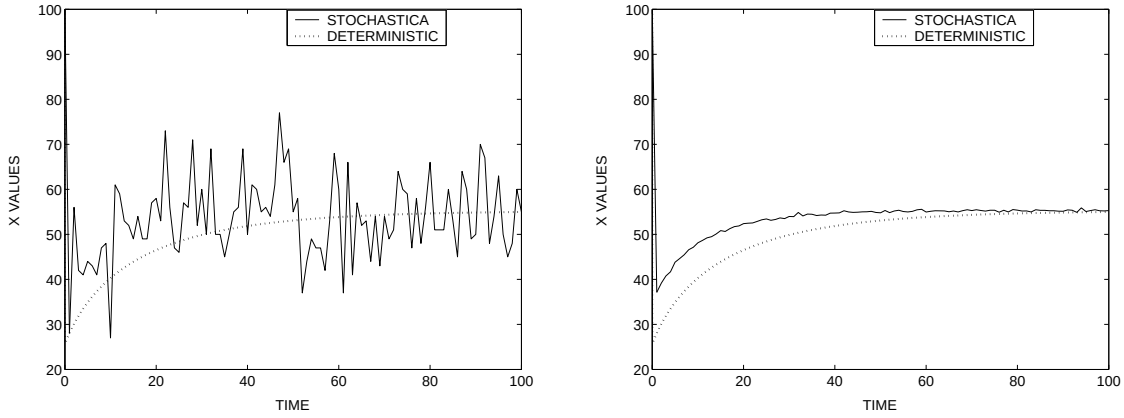
Table 4.15: Model Initial Conditions for Stochastica, ESS, and τ-Leaping

Trial	X	X_2	D	DX_2
1	100	500	23	0
2	0	2500	23	0
3	0	0	2	0
4	4	45	2	0
5	0	0	2	0
6	1	2	2	0
7	20	20	2	0

Table 4.16: Stochastica and ESS Mean Final Values, 1,000 Paths Used

		X	X_2	D	DX_2
Trial 1	Deterministic	54.9857	3022.2	0.0076	22.9924
	Stochastica	55.2227	1526.5	0.0143	22.9857
	ESS	55.0962	1523.1	0.0182	22.9817
	Matlab Direct	55.2020	1525.9	0.0149	22.9851
Trial 2	Deterministic	55.1571	3042.0	0.0076	22.9924
	Stochastica	55.2401	1527.3	0.0152	22.9848
	ESS	55.2028	1516.4	0.0172	22.9827
	Matlab Direct	55.2194	1528.3	0.0150	22.9850
Trial 3	Deterministic	0.0661	0.0087	1.9913	0.0087
	Stochastica	5.5313	32.1532	0.1718	1.8282
	ESS	5.8760	32.5493	0.1290	1.8710
	Matlab Direct	5.5986	32.5606	0.1469	1.8531
Trial 4	Deterministic	5.8708	68.9332	0.0564	1.9436
	Stochastica	5.6922	33.0207	0.1178	1.8822
	ESS	5.7960	32.6355	0.1390	1.8610
	Matlab Direct	5.6803	33.0533	0.1182	1.8818
Trial 5	Deterministic	0.0438	0.0038	1.9962	0.0038
	Stochastica	0.0472	0.0132	1.9927	0.0073
	ESS	0.9700	0.0058	1.9955	0.0045
	Matlab Direct	0.0438	0.0078	1.9955	0.0045
Trial 6	Deterministic	1.2873	3.3145	0.7527	1.2473
	Stochastica	0.0468	0.0115	1.9938	0.0062
	ESS	0.9620	0.0035	1.9962	0.0038
	Matlab Direct	0.0425	0.0041	1.9974	0.0026
Trial 7	Deterministic	1.3151	3.4655	0.7317	1.2683
	Stochastica	0.0435	0.0076	1.9948	0.0052
	ESS	0.9600	0.0013	1.9987	0.0013
	Matlab Direct	0.0431	0.0050	1.9966	0.0034

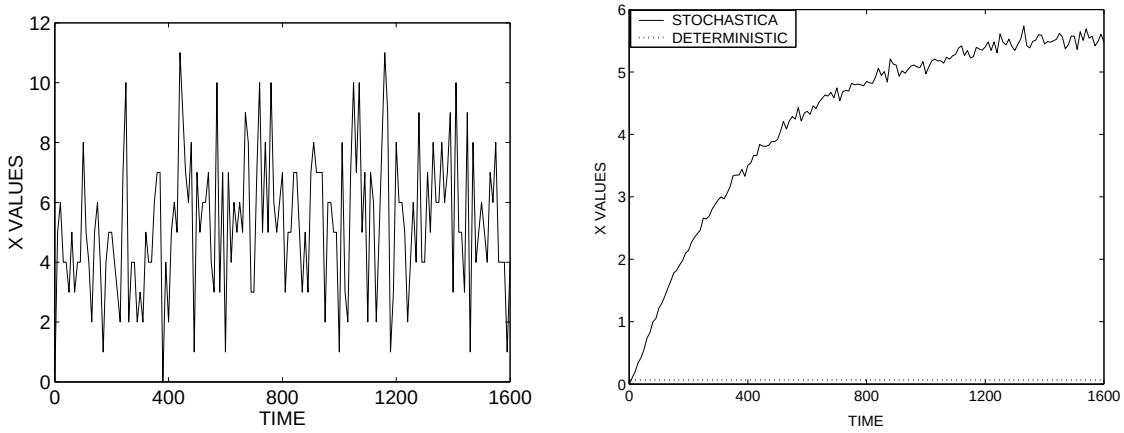
The final values for Stochastica and the deterministic problem for trials 1 and 2 are close for the X and DX_2 species. However, Stochastica's final value for X_2 is about half of the deterministic result for both trials and the final value for D is about twice that of the deterministic. The difference in values of D is not that significant since such small values are involved, but the difference is significant for the X_2 values. It is not clear why this difference occurred. Since there is only one real critical value, there is not another lower valued stable steady state to drag the mean final values down. The Matlab mean final values for both trials are very close to the Stochastica values. A single path and the path mean for the trial 1 calculations are shown in Figure 4.16.



(a) X values for a single path (b) X values for the mean path, 1,000 paths used

Figure 4.16: Stochastica trial 1 paths, paths are plotted at one time unit intervals

Rather than the 100 time units stated in section 3.3.1, trial 3 was run for 1600 time units because it took that long for the average X value to reach a somewhat steady state. The deterministic final values for trial 3 are the lower critical values while Stochastica's final values are close to the upper critical values except for the X_2 species. The final value for X_2 is about half that of the upper critical value. The deterministic and Stochastica final values for trial 4 are both close to the upper critical values. Again, the Stochastica final value for X_2 is about half that for the upper critical value. Unlike with trials 1 and 2, there is a lower critical value to



(a) X values for a single path (b) X values for the mean path, 1,000 paths used

Figure 4.17: Stochastica trial 3 paths, paths are plotted at 10 time unit intervals

drag the mean final value of X_2. However, the mean final X value does not show this effect. The Matlab mean final values for trials 3 and 4 are very close to the Stochastica values. It was not possible to compare deterministic and Stochastica results for initial values of X between the lower steady state and the unstable steady state because of the integer value restriction of Gillespie's method. A single path and the path mean for the trial 3 calculations are shown in Figure 4.17. Notice that the single path jumps around a lot.

For trial 5 the deterministic final values are the lower critical values and the Stochastica final values are close for all species. However, in trials 6 and 7 the deterministic final values are the upper critical values and the Stochastica final values are near the lower critical value. The Matlab mean final values for trials 5, 6, and 7 are very close to the Stochastica values. The Matlab X value for trial 5 is equal to the deterministic value. Deterministic and Stochastica results with initial values of X between the lower stable critical value and the the middle unstable critical value could not be studied because of the integer value restriction.

4.3.2 Exact Stochastic Simulator. The mean final values for the ESS calculations are presented in Table 4.16. The procedure is described in section 3.3.2

The final values for trials 1 and 2 are close to the deterministic values and those obtained using Stochastica and the Matlab implementation. As with Stochastica and the Matlab results, the final value for X_2 is about half that of the deterministic value.

Like the Stochastica and Matlab implementation results, the final results for trials 3 and 4 for ESS are near the upper critical values except for the X_2 species. The final value of X_2 is about half that of the upper critical value.

The ESS results for trials 5, 6, and 7 are close to the Stochastica and Matlab implementation results with the exception of the X species. Looking at an average of only twenty paths in ESS might not be enough here to declare definitively that there is a difference in the final averages from Stochastica, but as stated in section 4.3.2 it was not possible to calculate multiple paths at one time. However, calculating twenty paths with Stochastica and the Matlab implementation using the initial values for trials 5, 6, and 7 gave X values very close to the lower critical value. Thus the final values for the X value with ESS are significantly different than those calculated using Stochastica and the Matlab implementation.

4.3.3 τ-Leaping. Programming code written by Press et al. was used for the Poisson random number generator needed to carry out the calculations [29, pgs 297-299]. Care should be taken when implementing the τ-leaping method to ensure that leaps which would cause physically unrealistic negative values of any of the species are not carried out. Sampling the Poisson distribution to determine the number of times a reaction occurs may generate a larger number of reaction occurrences than the amount of a species actually allows for. As an example, the Hasty et al. protein model was solved using the trial 1 parameter values given in Table 4.13 with the following initial conditions: X = 50000, X_2 = 100, D = 23, DX_2 = 0. During the course of the calculations the species were at the following amounts: X = 694, X_2 = 24710, D = 1, DX_2 = 22 and a time step τ = 2.4353e-5 was calculated. For the reaction D + $X_2 \rightarrow DX_2$, the propensity function then equaled

24710 and the mean for the sampled Poisson distribution was 0.6018. The sampling indicated that the reaction occurred twice. However, this would cause the amount of D to be equal to -1, which is unrealistic. Note that this would not have occurred for the exact method since it steps through the reactions one at a time. Any time that the calculations indicated leaps that were not realistic the exact method was performed instead.

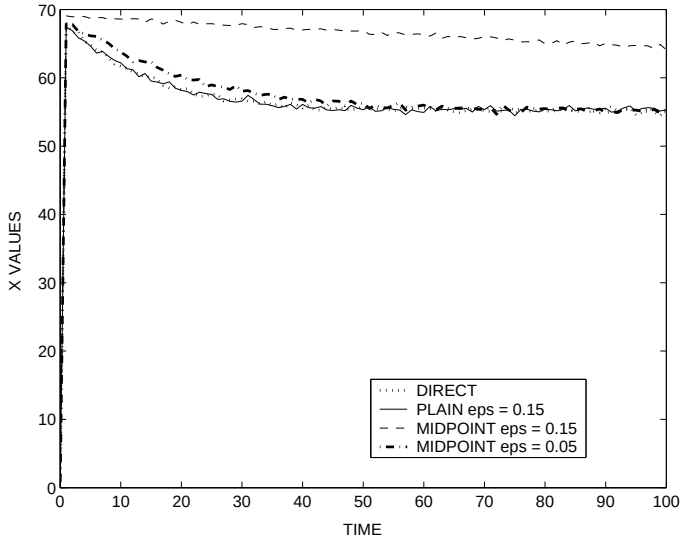


Figure 4.18: τ-Leaping: X Values For the Mean Path, Trial 2, 1,000 Paths Used

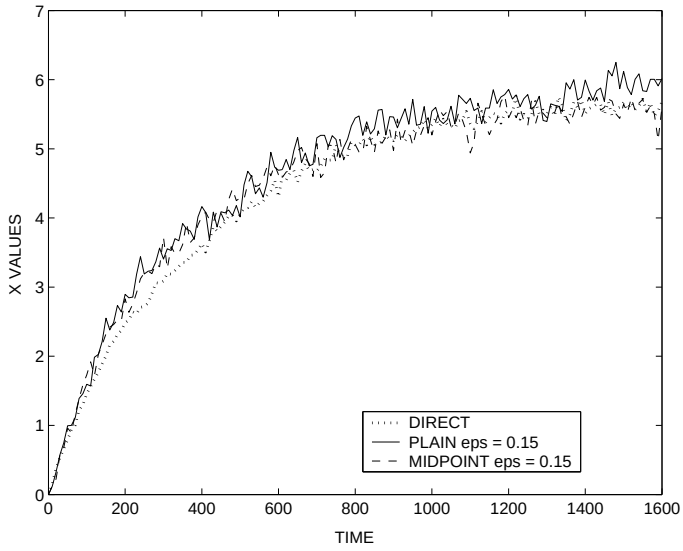


Figure 4.19: τ-Leaping: X Values For the Mean Path, Trial 3, 200 Paths Used

Table 4.17: τ-Leaping Mean Final Values, 1,000 Paths Used

			X	X_2	D	DX_2	Exact Reactions	Leaps
Trial 1	Direct Method		55.2020	1525.9	0.0149	22.9851	359366	-
	Plain τ-Leaping	$\epsilon = 0.05$	55.1450	1528.9	0.0147	22.9853	7976	26116
		$\epsilon = 0.15$	55.1929	1545.4	0.0183	22.9817	4416	11396
	Midpoint τ-Leaping	$\epsilon = 0.05$	55.0513	1515.3	0.0115	22.9885	11304	20086
		$\epsilon = 0.15$	50.7280	1262.7	0.0029	22.9971	5935	1914
Trial 2	Direct Method		55.2194	1528.3	0.0150	22.9850	407851	-
	Plain τ-Leaping	$\epsilon = 0.05$	55.2636	1532.5	0.0148	22.9852	7851	27177
		$\epsilon = 0.15$	55.2887	1548.3	0.0183	22.9817	4714	11597
	Midpoint τ-Leaping	$\epsilon = 0.05$	55.1566	1523.3	0.0111	22.9889	11799	20328
		$\epsilon = 0.15$	64.8725	2148.2	0.0008	22.9992	7230	660
Trial 3	Direct Method		5.5986	32.5606	0.1469	1.8531	545905	-
	Plain τ-Leaping	$\epsilon = 0.05$	5.7106	32.7822	0.1261	1.8739	377610	65459
		$\epsilon = 0.15$	5.9022	32.7660	0.1633	1.8368	161101	94919
	Midpoint τ-Leaping	$\epsilon = 0.05$	5.6037	32.0135	0.1434	1.8566	373635	60616
		$\epsilon = 0.15$	5.5698	29.1837	0.1840	1.8160	158694	77177
Trial 4	Direct Method		5.6803	33.0533	0.1182	1.8818	44672	-
	Plain τ-Leaping	$\epsilon = 0.05$	5.7395	33.0518	0.1172	1.8828	30461	5311
		$\epsilon = 0.15$	6.0062	33.7143	0.1306	1.8694	13048	7769
	Midpoint τ-Leaping	$\epsilon = 0.05$	5.6871	32.2374	0.1186	1.8814	30306	4940
		$\epsilon = 0.15$	5.7389	30.1194	0.1291	1.8709	12794	6312
Trial 5	Direct Method		0.0438	0.0078	1.9955	0.0045	111	-
	Plain τ-Leaping	$\epsilon = 0.05$	0.0448	0.0108	1.9949	0.0051	110	0
		$\epsilon = 0.15$	0.0447	0.0078	1.9953	0.0047	105	2
	Midpoint τ-Leaping	$\epsilon = 0.05$	0.0444	0.0088	1.9954	0.0046	112	0
		$\epsilon = 0.15$	0.0439	0.0063	1.9959	0.0041	105	2
Trial 6	Direct Method		0.0425	0.0041	1.9974	0.0026	230	-
	Plain τ-Leaping	$\epsilon = 0.05$	0.0440	0.0062	1.9965	0.0035	232	2
		$\epsilon = 0.15$	0.0444	0.0069	1.9960	0.0040	205	11
	Midpoint τ-Leaping	$\epsilon = 0.05$	0.0437	0.0045	1.9970	0.0030	231	2
		$\epsilon = 0.15$	0.0452	0.0085	1.9949	0.0051	194	8
Trial 7	Direct Method		0.0431	0.0050	1.9966	0.0034	1008	-
	Plain τ-Leaping	$\epsilon = 0.05$	0.0449	0.0075	1.9957	0.0043	1002	6
		$\epsilon = 0.15$	0.0469	0.0093	1.9944	0.0056	561	137
	Midpoint τ-Leaping	$\epsilon = 0.05$	0.0462	0.0118	1.9937	0.0063	1001	6
		$\epsilon = 0.15$	0.0460	0.0094	1.9946	0.0054	564	129

Because the calculations using the parameters and initial conditions for trial 3 were taking a very long time, only 200 paths were calculated using the τ-leaping method as opposed to the 1000 paths stated in the procedure in section 3.3.3. The mean final values for the direct method and the plain-τ leaping method along with the mean number of exact reactions and mean number of leaps performed are presented in Table 4.17.

The plain method for both ϵ values and both trials 1 and 2 drastically reduced the number of exact reactions that needed to be calculated while still obtaining final mean values very close to the direct method results. The direct method performed about 360,000 reactions for trial 1 and 408,000 reactions for trial 2. Using an ϵ of 0.05, the plain method reduced the number of exact reactions to about 8,000 for both trials and carried out about 27,000 leaps. Increasing ϵ to 0.15 roughly halved the number of mean exact reactions and leaps that were performed for both trials when using an ϵ of 0.05. The final values for the midpoint method using an ϵ of 0.05 are very close to the direct method values for all species. Roughly 4,000 more exact reactions were performed than for the plain method with an ϵ of 0.05 for both trials, but about 7,000 fewer leaps were required. The number of leaps performed for the midpoint method using an ϵ of 0.15, were much fewer than for the other calculations, but the values obtained are significantly different from the direct method values for species X and X_2. The mean paths for the trial 2 calculations are shown in Figure 4.18. The mean path for the plain method using an ϵ of 0.15 lies on top of the mean path for the direct method. The mean path for the midpoint method using an ϵ of 0.05 is also a close approximation for the direct method path. However, when using an ϵ of 0.15, the midpoint path is quite different.

For trial 3 the direct method performed around 545,000 mean reactions. The mean values for the plain method using both ϵ values are very close to the direct method values for all species. The number of exact reactions were reduced by about 170,000 using an ϵ of 0.05 and about 384,000 ϵ of 0.15. But about 65,000 and 95,000

leaps were required when using ϵ's of 0.05 and 0.15. Since a leap calculation takes more computational time than an exact reaction, this is most likely the reason for the long calculation times. The final values for the midpoint method when using an ϵ of 0.05 are all very close to the direct method values, but the values for the X_2 and D species when using an ϵ of 0.15 are significantly different in terms of relative error. About 168,000 exact reactions and 65,000 leaps are performed with an ϵ of 0.05. Using an ϵ of 0.15 reduces the number of exact reactions to about 158,000 but increases the number of leaps to around 77,000. As stated about the plain method, because the leaps takes more computational time than an exact reaction, this did not translate into a lot of computational savings time. The mean paths for the direct method and plain and midpoint τ-leaping methods with an ϵ of 0.15 for the trial 3 calculations are shown in Figure 4.19. The τ-leaping mean paths appear to be good approximations to the direct mean path.

The direct method simulated about 45,000 reactions for trial 4. The plain and midpoint methods reduced the number of exact reactions to about 30,000 and performed around 5,000 leaps and obtained values very close to the direct method. Clearly this is not a dramatic improvement. When ϵ was increased to 0.15, the number of exact reactions reduced to about 13,000 for the plain and midpoint methods. The number of leaps was reduced to about 8,000 for the plain method and about 6,000 for the midpoint method. The midpoint value for species X_2 with and ϵ of 0.15 differs significantly from the direct method value.

For trials 5, 6, and 7 the plain and midpoint τ-leaping mean values are all very close to the direct method values. Note that for trials 5 and 6 very few leaps are performed. The τ-leaping is not providing any real benefit because of the low value values. For trial 6, using an ϵ of 0.15 does cut in half the number of exact reactions and increase the number of leaps for both the plain and midpoint methods. This is because the initial values of the X and X_2 species has now increased from trials 5 and 6 to 20.

For this model, the plain τ-leaping method with an ϵ of 0.15, gives good accuracy when compared to the direct method and results in a good reduction of the number of reactions that are simulated.

4.4 UNC

The model initial conditions used for the UNC calculations are given in Table 4.18. The results are presented in two tables. Table 4.19 contains the mean final values for the proteins and the messenger RNA's. Table 4.20 contains the mean final values for the DNA sites both when they are and are not bound to protein A. The backward Euler form given by equation (4.8) was used for the Euler-Maruyama calculations. The procedure is described in section 3.4.

Table 4.18: Model Initial Conditions used for the UNC Calculations

A	R	mRNA.A	mRNA.R	AR	DA	DR	DAA	DRA
0	0	0	0	0	50	50	0	0

Table 4.19: UNC Protein Mean Final Values, 20 Paths Used

		A	R	mRNA.A	mRNA.R	AR
deterministic	ode23s	0.0337	186694.9	251.4823	39.5698	12572.8
	backward Euler	0.0333	188984.8	251.4644	39.9905	12572.0
stochastic	UNC - fully continuous	0.0380	169208.2	254.8613	35.4570	12806.1
	Euler-Maruyama	0.0309	195329.0	241.5760	41.8331	12063.3
	UNC - fully discrete	0.4300	189854.9	250.1075	42.5400	12576.9
	Stochastica	0.0300	189696.5	250.8875	39.6400	12578.4
	UNC - hybrid	0.0353	175582.8	250.4183	31.6795	12488.5

The ode23s and backward Euler results are very close for all species. The UNC fully discrete mean final value for species A is the only stochastic value that is significantly different from the deterministic values. The values for species R are close to the deterministic with the UNC fully continuous value as the farthest away. The EM value for mRNA.A is the farthest from the deterministic value with all the stochastic mean values being close to the deterministic. The UNC fully continuous and hybrid

Table 4.20: UNC DNA Mean Final Values, 20 Paths Used

		DA	DR	DAA	DRA
deterministic	ode23s	49.9665	49.9832	0.0335	0.0168
	backward Euler	49.9669	49.9815	0.0331	0.0166
stochastic	UNC - fully continuous	49.9472	49.8527	0.0528	0.1473
	Euler-Maruyama	48.1100	49.3562	0.0170	0.0004
	UNC - fully discrete	49.9775	49.9925	0.0225	0.0075
	Stochastica	49.9725	49.9775	0.0275	0.0225
	UNC - hybrid	49.9650	49.9800	0.0350	0.0200

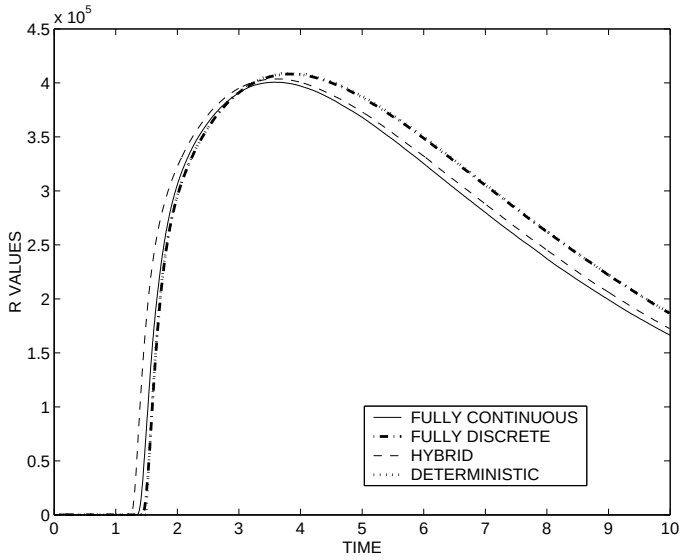


Figure 4.20: UNC: R Values for the Mean Path, 20 Paths Used

values are significantly different than the deterministic for species mRNA.R, with about 10% and 20% relative errors respectively. The stochastic final values for all the approaches are close to the deterministic value for species AR with the EM value being the farthest away. All the final values for the stochastic approaches are close to the deterministic values for species DA and DR. For species DRA, the Stochastica and UNC hybrid values are close to the deterministic value. The UNC continuous value is almost ten times as much while the UNC fully discrete value is about half the deterministic value and the UNC fully discrete value is almost zero. Figure 4.20 shows the mean paths calculated using the methods in the UNC software in compar-

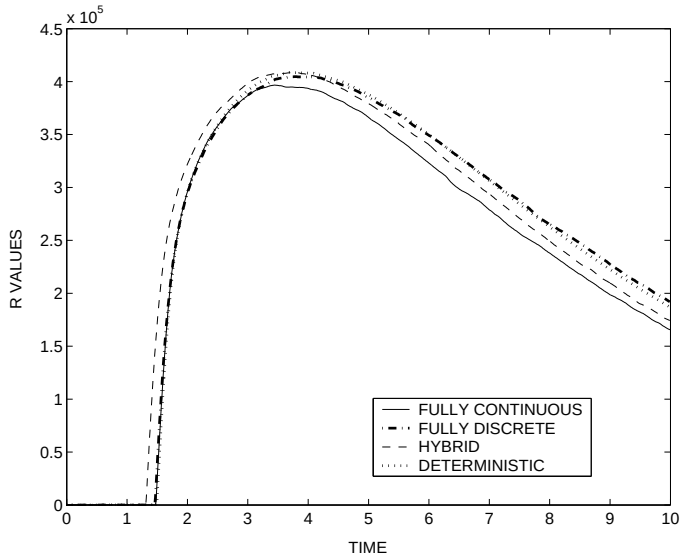


Figure 4.21: UNC: R Values for a Single Path

ison to the deterministic path for the R protein. Note that the fully discrete path lies on top of the deterministic path. Figure 4.21 shows a single path calculated using the methods in the UNC software in comparison to the deterministic path for the R protein.

4.5 Summary

The single differential equation (2.10) with additive noise of the form g(x) $= \sqrt{\sigma^2}$ and multiplicative noise of the form g(x) = x$\sqrt{\sigma^2}$, was examined with the explicit Euler-Maruyama method. For the larger σ^2 values, given in Table 4.4, the majority of final X values for additive noise tended to end up near the upper critical value while the majority of final X values for multiplicative noise tended to end up near the upper lower critical value. The multiplicative noise form was also solved using the explicit Milstein method. The results were very close to those obtained using the EM method.

The system of differential equations (2.6) through (2.9) with with diagonal additive noise of the form $g^{k,k}(x) = \sqrt{\sigma^2}$ and diagonal multiplicative noise of the

form $g^{k,k}(x) = x^k\sqrt{\sigma^2}$ was examined with the implicit Euler-Maruyama method. The larger σ^2 values with the additive noise had a stronger effect on the system than with the multiplicative noise. The multiplicative noise form was also solved using the implicit Milstein method. The results were very close to those obtained using the EM method.

The order of weak convergence for the EM method was investigated using error plots for the single SDE with the previously described additive and multiplicative noise. Using the deterministic value for the true value with additive noise required decreasing the σ^2 value to 0.0001. Using a path created with small Δt increments to approximate the true value resulted in approximated orders of convergence close to the theoretical order of convergence of 1 when using $\sigma^2 = 1$ for additive noise and approximated values near 1 when using $\sigma^2 = 0.05$ for multiplicative noise.

The order of strong convergence for the EM method was investigated using error plots for the single SDE with the previously described additive and multiplicative noise. The sum of the deterministic value and $\sqrt{\sigma^2}$W(T), where T is the final time, was used as the true value for the additive noise. This was not successful as all of the σ^2 values resulted in approximated orders of convergence of 0. Using a path created with small Δt increments to approximate the true value resulted in approximated orders of convergence close to the theoretical order of convergence of 0.5 when using $\sigma^2 = 1$ for additive and multiplicative noise. Good results were also obtained when using $\sigma^2 = 1$ for the Milstein method with multiplicative noise, as approximated orders of convergence near the theoretical value of 1 were calculated.

The BioSPICE modules Stochastica and Exact Stochastic Simulator, which implement Gillespie's exact method, were used to examine the full system of differential equations, (2.6) through (2.9). A Matlab implementation of Gillespie's direct method was also used. Sometimes the mean path for Gillespie's method had some of the species values near the deterministic values for the same initial conditions while sometimes the mean values for all the species would be different than the de-

terministic values. The results were not consistent for every different set of model parameters.

The direct method and plain and midpoint τ-leaping methods were implemented in Matlab and the number of reactions simulated were compared. The results depended on the set of model parameters used. A couple of the experiments showed that the τ-leaping method could drastically reduce the number of simulations. For experiments with very small species' values, very few if any leaps were performed. Sometimes so many leaps were performed that while the number of simulations were fewer than with the direct method, the time of calculation was longer.

The BioSPICE modules Gene Regulatory Analysis and Stochastica Simulator and a University of North Carolina contribution were examined. However, additional information concerning how the calculations are performed is needed before they can be replicated in Matlab.

V. Conclusions

The goal of this document was to explore methods used to add randomness to models of biochemical reactions and then understand approaches for solving these stochastic models. These methods treat the quantities of the species as random variables. The different approaches can be broken into two categories, those that treat the state space as continuous and those that treat it as discrete. The continuous approaches, the explicit and implicit Euler-Maruyama methods and the explicit and implicit Milstein methods, involve solving stochastic differential equations, while the discrete approaches, Gillespie's exact method and his plain and midpoint τ-leaping methods, perform random walks for a Markov process.

The obvious question is, which method is best? It is not clear that a method can be determined to be the 'best' method. If an investigator wants to compute only one random path for his model, then clearly whether or not his species' quantities are only allowed to take on nonnegative integer values or if they can be continuous values, will decide his choice of using one of the continuous methods or one of Gillespie's methods. However, if the investigator is interested in looking at a mean path, then the choice is not as clear-cut because Gillespie's methods produce non-integer values for the mean path.

All of the methods that were examined in this document were fairly simple. They were easy to understand and easy to implement in Matlab. The major difference between the stochastic differential equation methods and Gillespie's methods, other than the state spaces of the species' random variables, is how the randomness is added into the model. The Euler-Maruyama and Milstein methods have an actual randomly generated noise term whose mean and variance can be manipulated as desired. Gillespie's methods however have no such term. The randomness is inherent in the model and is influenced by the quantities of the different species involved.

If one of the continuous methods are chosen, the question becomes what form should the noise term be? It would seem that the model being investigated would dictate what noise is too much and what noise is not enough. The Hasty et al. protein regulation model that was studied here had very small critical values for the parameters that were used. Noise that was considered extremely large for this model might not even be enough to be noticed for a system that had much larger critical values.

The differences between the performances of Gillespies exact method and his τ-leaping methods also indicate that the results heavily depend on the model. Different sets of parameter values used for the Hasty et al. protein model resulted in different levels of effectiveness of the τ-leaping methods when compared to the exact method.

In the end, the method that an investigator should choose appears to depend very much on the model being studied and perhaps on the type of approach the investigator is most comfortable with.

The work done in this thesis should be considered as a basic foray into the area of stochastic modeling. It is not meant to be a complete and final statement. It is hoped that it would provide a good learning tool for someone desiring to get into this area.

5.1 Recommendations

There are many different paths for future work. During the course of the work presented in this document, a couple of journal articles that presented methods for variable step size evaluation of stochastic differential equations were found. It would be interesting to study the improvements gained from using a variable step size method as opposed to the fixed step size methods examined in this document. More complex noise than the simple diagonal noise used for the full system could be studied. It would be necessary to understand the methods for numerically approximating multiple stochastic integrals, which could prove to be a significant area

of study itself. Looking at methods of higher convergence order could also be interesting. Kloeden and Platen [24] give several numerical schemes of higher order, such as Runge-Kutta methods. Again, this would lead into the study of numerically approximating multiple stochastic integrals. The differing properties of the Ito and Strotanovich integrals could also be examined. The better than expected strong order of convergence results for the Euler-Maruyama method with additive noise could be investigated. Also, solving for critical values of multi-dimensional systems, as stated in the GRASS results section 4.2.4, could be explored. This would involve optimization techniques.

Gillespie's methods could be studied more in depth to gain insight into why the mean discrete solutions arrive at somewhat steady states that have significantly different values for some species when compared to the deterministic steady states. Gillespie has also created an implicit τ-leaping method that could be examined. The UNC module had an intriguing idea of mixing the continuous and discrete approaches. More exploration could be done in this regard.

Appendix A. Appendix

A.1 Stochastica

This is the Stochastica input file used for trial 1.

```
//Hasty 1
-> X                                .04
2*X -> Y                            1
Y -> 2*X                            1
D + Y -> DY                         1
DY -> D + Y                         1
DY + P -> DY + P + 3*X              4
X ->                                10
***
X                   100
Y                   500
D                   23
DY                  0
P                   2
***
DURATION                            100
STEP                                .1
MAX_RUNS                            1000
SPECIES_WATCHED                     X Y D DY
***
```

A.2 GRASS

This is the input file that was used for the GRASS program.

```
thesis_run  # Run name (file names will be built with this as a root, eg. Name.init,
Name.reactions)

0.0001      # Time step
50          # Run duration
0.01        # File report interval (how often to output to file)
1           # Random number generator seed (-ve = seed from clock)
```

1 # Noise flag (0 = deterministic, 1 = normal SDE)
2 # Number of genes (N)
50 # Number of copies of each gene (m) (plasmid copy number, for example; set m=1 for genomic DNA)

0.1 # epsilon - scaling factor for equilibrium constant rates: for reaction with eq. const. K, forward rate = K/eqps, reverse = 1/eps (fwd/rev = K)

N x 1 - Initial conditions for free monomers: P(1,1), P(2,1) ...

0 0

N x 1 - Initial conditions for free dimers: P(1,2), P(2,2) ...

0 0

N x 1 - beta: base production rates for each monomer (ie. rate of production from bare DNA, no regulation)

5 4

N x 1 - gamma: degradation rates for each monomer species

0.7 0.5

N x 1 - gamma2: degradation rates for each dimer species (if it exists)
– [negative value] –> set same as monomer degradation rate

-1 -1

N x 1 - K2: dimerization constants for each dimer – K2(i)=0 –> species i has no dimeric form

0.05 0.075

N x 1 - Ta: transcriptionally active form of each protein - 1 = monomer, 2 = dimer

2 2

N x N - Kdna: gene-protein association constants - Kd(i,j)[row,col] = association between gene i and protein j (which form of protein is set by Ta(j))

0.5 0.5
0.5 0.5

N x N - alpha: gene regulation constants - alpha(i,j)[row,col] = regulation of gene i by protein j (which form of protein is set by Ta(j))

0.25 1.25
0 1

Bibliography

1. Apostol, Tom M. *Mathematical Analysis* (2nd Edition). Massachusetts: Addison-Wesley Publishing Company, 1974

2. Bharucha-Reid, A.T. *Elements of the Theory of Markov Processes and Their Applications*. New York: Dover Publications, Inc., 1998

3. BioCharon. "Modeling and simulation of bioregulatory networks in BIOCHARON". 4 September 2003 https://community.biospice.org/

4. Beckwith, David, "Numerical Realizations of the Chemical Master Equation with Reaction Coefficient". Help Manual included with Stochastica. 4 September 2003 https://community.biospice.org/

5. BioSPICE. 4 September 2003 https://community.biospice.org/

6. Boyce, William E.; DiPrima, Richard C. *Elementary Differential Equations* (6th Edition). New York: John Wiley & Sons, Inc., 1997

7. Campbell, Matthew. *Cell Modeling*. MS thesis, Graduate School of Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB OH, March 2002. AFIT/GAM/ENC/02M-01.

8. Defense Advanced Research Projects Agency. Project Proposal Pamphelet. 4 September 2003 http://www.darpa.mil/ipto/Solicitations/PIP_01-26.html

9. Datta, Biswa Nath. *Numerical Linear Algebra and Applications*. California: Brooks/Cole Publishing Company, 1995

10. Exact Stochastic Simulator. 4 September 2003 http://biocomp.ece.utk.edu

11. Gardner, Timothy S.; Cantor, Charles R.; Collins, James J. "Construction of a Genetic Toggle Switch in Escherichia Coli," *Nature* , 403:339-342 (2000)

12. Gepasi. 4 September 2003 http://www.gepasi.org/

13. Gillespie, Daniel T. "A General Method for Numerically Simulating the Stochastic Time Evolution of Coupled Chemical Reactions," *Journal of Computational Physics*, 22:403-434 (1976)

14. Gillespie, Daniel T. "Exact Stochastic Simulation of Coupled Chemical Reactions," *The Journal of Physical Chemistry*, 81:2340-2361 (1977)

15. Gillespie, Daniel T., "Approximate Accelerated Stochastic Simulation of Chemically Reacting Systems," *Journal of Chemical Physics*, 115:1716-1733 (2001)

16. Grigoriu, Mircea. *Stochastic Calculus: Applications in Science and Engineering*. Boston: Birkhäuser 2002

17. Hasty, Jeff; Pradines, Joel; Dolnik, Miles; Collins, J. J. "Stochastic Regulation of Gene Expression," *American Institute of Physics Conference Proceedings*, 501:191-196 (2000)

18. Hasty, Jeff; Pradines, Joel; Dolnik, Miles; Collins, J. J. "Noise-Based Switches and Amplifiers for Gene Expression," *Proceedings of the National Academy of Science*, 97:2075-2080 (2000)

19. Heinrich, R.;Rapoport, S. M.; Rapoport, T. A. "Metabolic Regulation and Mathematical Models," *Progress in Biophysics and Molecular Biology*, 32:1-82 (1977)

20. Higham, Desmond J. "An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations," *SIAM Review*, 43(no.2):525-546

21. Honerkamp, Josef. *Stochastic Dynamical Systems: Concepts, Numerical Methods, Data Analysis*. New York: VCH Publishers, Inc., 1994

22. JigCell. 4 September 2003 http://gnida.cs.vt.edu/ cellcyclepse/

23. Kulkarni, Vidyahar G. *Modeling and Analysis of Stochastic Systems*. London: Chapman & Hall, 1995

24. Kloeden, Peter E.; Platen, Eckhard. *Numerical Solution of Stochastic Differential Equations*. New York: Springer-Verlag, 1995

25. McAdams, Harley H.; Shapiro, Lucy. "Circuit Simulation of Genetic Networks," *Science*, 269:650-656 (1995)

26. McAdams, Harley H.; Arkin Adam. "It's a Noisy Business! Genetic Regulation at the Nanomolar Scale," *Trends in Genetics*, 15(2):65-69 (1999)

27. Mendes, Pedro; Kell, Douglas B. "Non-Linear Optimization of Biochemical Pathways: Applications to Metabolic Engineering and Parameter Estimation," *Bioinformatics*, 14(10):869-883 (1998)

28. Neter, John; Kutner, Michael H.; Nachtsheim, Christopher J.; Wasserman, William. *Applied Linear Statistical Models* (4th Edition). Massachusetts: WCB/McGraw-Hill, 1996

29. Press, William; Teukolsky, Saul; Vetterling, William; Flannery, Brian. *Numerical Recipes in C++: The Art of Scientific Computing* (2nd Edition). New York: Cambridge University Press, 2002

30. Smolen, Paul; Baxter, Douglas A.; Byrne, John H. "Modeling Transcriptional Control in Gene Networks: Methods, Recent Results, and Future Directions," *Bulletin of Mathematical Biology*, 62:247-292 (2000)

31. Stakgold, Ivar. *Green's Functions and Boundary Value Problems* (2nd Edition). New York: John Wiley & Sons, Inc., 1998

32. Tortora, Gerard S.; Funke, Berdell R.; Case, Christine L. *Microbiology: An Introduction*. California: Addison Wesley Longman, Inc., 1998

33. Wackerly, Dennis D.; Mendenhall, William; Scheaffer, Richard L. *Mathematical Statistics With Applications* (6th Edition). California: Duxbury, 2002

34. Young Jaqueline B. *Deterministic Intra-Cellular Modeling*. MS thesis, Graduate School of Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB OH, March 2003. AFIT/GCS/ENC/03M-1.

Vita

Thomas Hopkins was born in Fort Dix, New Jersey. As a young child, he lived in Germany and Oklahoma before settling in Springdale, Arkansas. He graduated from Springdale High School in 1992. He obtained Bachelor's of Science degrees in Math and Chemistry from the University of Arkansas at Fayetteville in 2001.

From August 2001 to September 2003 he was employed by Dr. Quinn as a Research Assistant while working towards the completion of a Master's degree in Applied Mathematics.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
09-30-2003	**Master's Thesis**	Aug 2001 – Sept 2003

4. TITLE AND SUBTITLE

Stochastic Intra-Cellular Modeling

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Hopkins, Thomas E., Civilian

5d. PROJECT NUMBER
ENR #2003-017

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 P Street, Building 640
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GAM/ENC/03S-1

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFOSR
Attn: Dr. Kozumbo
801 N. Randolph
WPAFB OH 45433-7765
Phone: (703) 696-7720

DAGSI
Attn: Dr. Downie
3155 Research Blvd., Suite. 205
Kettering, OH 45420
Phone: (937) 971-4000

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Air Force personnel may sometimes come into contact with potentially harmful chemicals while performing their duties. Of course the Air Force desires to keep any potential health risks to its members to a minimum. To this end the Air Force would like to identify which chemicals are toxic, their level of toxicity, and the processes by which these chemicals disrupt normal biological activities at the cellular level. The development of mathematical models can be of great benefit to toxicity studies.

Because real world systems involve randomness, that is noise, and the desire is to create mathematical models to represent those systems, it is necessary to study approaches used to add noise to mathematical models. This document examines different methods for incorporating noise into biochemical systems. The various quantities involved in the reactions are treated as random variables. The methods can be separated into two categories: those which treat the random variable as having a continuous state space and those which treat the random variable as having a discrete state space. These different approaches are compared in order to better understand what type of method would be best used for adding noise to a model and how the model is affected.

15. SUBJECT TERMS

Mathematical Modeling, Stochastic Intracellular Modeling, Stochastic Differential Equations, Poisson Process, Fokker-Planck, Markov Process, Tau-Leaping, Euler-Maruyama Method, Milstein Method

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dennis W. Quinn, ENC
U	**U**	**U**	**UU**	113	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-6565, ext 4522; e-mail: Dennis.Quinn@afit.edu

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18